UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UM SOFTWARE PARA RESOLVER O PROBLEMA QUADRÁTICO LINEAR DETERMÍNISTICO DISCRETO DE CONTROLE OTIMO

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

THOMAS WALTER MCCUTCHEN

FLORIANÓPOLIS

SANTA CATARINA - BRASIL

AGOSTO - 1990

UM SOFTWARE PARA RESOLVER O PROBLEMA QUADRÁTICO LINEAR DETERMINÍSTICO DISCRETO DE CONTROLE ÓTIMO

THOMAS W. MCCUTCHEN

Esta dissertação foi julgada adequada para a obtenção do título de

"MESTRE EM ENGENHARIA"

Especialidade Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-Graduação.

Prof. Robert Wayne Samohyl, Ph.D.
Orientador e Presidente da Banca

Banca Examinadora:

Prof. Edgar Augusto Lanzer, Ph.D.
Co-orientador

Prof. Ricardo Miranda Barcia, Ph.D.

Prof. Sérgio Fernando Mayerle, M. Eng.

AGRADECIMENTOS

Gostaria de expressar os meus agredecimentos:

- Aos Professores Bob e Lanzer pela orientação do trabalho e por ter aguentado o meu Português escrito.

- Ao Professor Barcia, especialmente pelos seus esforços na obtenção do meu visto de estudante.

- Ao Professor Sérgio Mayerle, pelas suas sugestões, e por ter participado na banca examinadora.

- Ao Professor David Kendrick da University of Texas, por ter mandado uma cópia do seu livro e o programa que o seu equipe desenvolveu.

- Ao Destre e Fábio, que me ajudaram na resolução do alguns problemas de Turbo Pascal.

- Ao contribuinte brasileiro, que através da CAPES me deu apoio financeiro.

RESUMO

Este trabalho consiste em um software que resolve o problema quadrático linear determinístico discreto (QLDD) de controle ótimo.

No problema QLDD otimiza-se uma função quadrática (1) através do tempo sujeita a um conjunto de equações de sistema lineares (2). Não existe incerteza, e considera-se o tempo em intervalos discretos. A forma padrão é:

Achar u_k para $k = 0, 1, \ldots, N-1$

$$(1)\quad \text{Min } J = \tfrac{1}{2}x'_N W_N x_N + w'_N x_N + \sum_{k=0}^{N-1} \left(\tfrac{1}{2}x'_k W_k x_k + w'_k x_k + \tfrac{1}{2}u'_k L_k u_k + l'_k u_k\right)$$

Sujeito a:

$$(2)\quad x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

Pode-se influenciar os vetores de estado (x_k) através da seleção dos vetores de controle (u_k), que estão ligados através de (2). Os vetores exógenos (z_k) são conhecidos, porém fora do controle do analista.

Escolhe-se os valores de u_k, e os consequêntes valores de x_k, de maneira que a função objetivo (1) seja minimizada. Na (1), W_k, w_k, L_k e l_k são matrizes de penalidades (custos). Na (2), A, B, C e D são matrizes de constantes.

Trata-se também do caso de "tracking", em que especifica-se valores desejados para x_k e u_k. Então, penaliza-se afastamentos dos valores ideais, e não os valores absolutos.

Decreve-se equações de maior ordem, onde x_{k+1} não depende apenas do período k, mas também de períodos anteriores. Explica-se os tipos de multiplicadores, e deriva-se algoritmos de solução.

Descreve-se duas aplicações de controle ótimo, e em apêndice é apresentado o manual do software.

ABSTRACT

This work is a software package that solves quadratic linear deterministic discrete (QLDD) problems of optimal control.

In the QLDD problem we desire to optimize a quadratic function (1) through time subject to a set of linear system equations (2). We assume no uncertainty, and time is divided in discrete intervals. The standard form is:

Find u_k for $k = 0, 1, \ldots, N-1$

(1) $$\text{Min } J = \tfrac{1}{2}x_N' W_N x_N + w_N' x_N + \sum_{k=0}^{N-1} \left(\tfrac{1}{2}x_k' W_k x_k + w_k' x_k + \tfrac{1}{2}u_k' L_k u_k + l_k' u_k\right)$$

Suject to:

(2) $$x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

for $k = 0, 1, \ldots, N-1$

with x_0 given.

The state vectors (x_k) may be influenced though the selection of control vectors (u_k), which are linked though (2). Exogenous vectors (z_k) may not be controlled, but are considered known.

The values of u_k, and the consequent values of x_k are chosen to minimize the objective function (1). In (1), W_k, w_k, L_k and l_k are penalty (cost) matrices. In (2), A, B, C and D are constant matrices.

The tracking case is also treated, in which target values are specified for x_k and u_k. In this case, variations from the ideal values are penalized, and not absolute values.

Higher order equations are also examined, in which x_{k+1} does not depend only on period k but also on previous periods. The multipliers are explained, and solution algorithms are derived.

Two applications of optimal control are described, and a manual for the software package is furnished.

ÍNDICE

CAPÍTULO 1: INTRODUÇÃO:

1.1 Descrição do trabalho:

Este trabalho consiste no desenvolvimento de um software escrito em Turbo Pascal 5.0 para a resolução de problemas quadráticos lineares determinísticos discretos (QLDD) de controle ótimo. O nome deste software é PQL, para Programação Quadrática Linear.

As habilidades do software são:

1. Tratamento do problema geral e do caso de "tracking".

2. Controle do número de algarismos na saída.

3. Saída facultativa das matrizes intermediárias calculadas na solução do problema.

4. Inclusão facultativa de vetores exógenos.

5. Pacote gráfico acompanhante que mostra os valores da solução. No caso "tracking" também mostra os valores desejados.

6. Cálculo e saída facultativos das matrizes dos multiplicadores.

7. Identificação de problemas que não têm solução porque uma das matrizes que o algoritmo manda invertir não é inversível.

8. Entrada de dados através do teclado ou de um arquivo de dados. No primeiro caso, um arquivo de dados é criado para uso subsequente.

9. Opções de verificar e modificar os dados durante a execução do programa.

10. Mensagens na tela que relatam o progresso do programa.

11. Identificação facultativa dos elementos dos vetores com nomes significativos.

1.2 Justificativa do trabalho:

Até agora, não se sabe de um programa com comentários e documentação em Português que resolve este tipo de problema. Um pacote escrito em Fortran chamado QLPWIN está sendo aperfeiçoado nos Estados Unidos na University of Texas. Porém, até o início deste trabalho não existia documentação completa em inglês para o pacote. Além disso, o QLPWIN só oferecia as primeiras seis habilidades citadas na seção anterior.

Os arquivos do tipo texto do PQL podem ser alterados e recompilados, o que implica várias vantagens sobre os arquivos objetos do QLPWIN. A flexibilidade de fazer modificações pequenas e simples dá um grande aumento na utilidade do programa. Por exemplo, com poucas mudanças o programa pode ser alterado para tratar o caso do vetor de controle com impacto imediato, além do caso do vetor de contróle com impacto defasado. De forma similar, em lugar de ter limites fixos para o número de períodos e nas dimensões dos vários tipos de vetores, a memória pode ser redistribuída entre eles de acordo com o problema.

Outras mudanças ou adições poderiam ser feitas no programa para adaptá-lo a um problema específico. As vezes, é necessário fazer cálculos preliminares para obter os dados necessários para resolução de um problema usando o algoritmo de controle ótimo QLDD. Tais cálculos podem ser extensos e complexos, mas podem ser feitos em rotinas especiais e ligados ao programa principal.

Finalmente, o PQL pode servir como uma base para os problemas de controle ótimo mais complexos. Os arquivos de texto podem ser modificados para tratar problemas estocásticos ou com equações de sistema não-lineares.

1.3 Resumo dos capítulos:

No Capítulo 2, apresenta-se um resumo da literatura, principalmente os livros de Gregory Chow (1975) e do David Kendrick (1982). Explica-se o problema geral de controle ótimo, e o caso sendo tratado aqui, o QLDD. Explica-se também o método de programação dinâmica. Usa-se este método no desenvolvimento do algoritmo usado pelo software. A seguir, trata-se o caso especial do problema de "tracking", e o tratamento de vetores defasados de estado e controle. Finalmente, explica-se os vários tipos de multiplicadores.

No Capítulo 3 apresenta-se um desenvolvimento, passo a passo, do algoritmo usado para a solução do problema QLDD. Trata-se tanto o caso do vetor de controle defasado quanto o caso do vetor de controle imediato.

O Capítulo 4 fornece duas aplicações do algoritmo, resolvidas pelo software. A primeira aplicação é um problema micro-econômico, de controle de produção e admissão de empregados. A segunda é uma aplicação macro-econômica, em que tenta-se determinar as condições internacionais mais favoráveis para reduzir a dívida externa e provocar crescimento na economia brasileira. Embora este segundo exemplo não produziu resultados econômicos muito relevantes, ainda é muito útil no sentido de ilustrar como o pesquisador poderia proceder frente de dificuldades muito comuns em controle ótimo.

No primeiro apêndice é apresentado o manual do PQL. Esta seção é autosuficiente, e inclui-se um resumo do problema QLDD. Além deste resumo, este apêndice pode ser separado em duas partes: Na primeira parte trata-se apenas do uso do software, sendo principalmente uma descrição do formato exigido do arquivo de dados que o programa lê. A segunda parte fornece informações mais detalhadas sobre as unidades que compõem o software. Esta parte é mais orientada para pessoas que querem modificar o programa para adaptá-lo a um problema particular, ou para um tipo de problema semelhante.

No segundo apêndice é dada a forma formal do algoritmo para problemas QLDD.

CAPÍTULO 2: RESUMO DA LITERATURA:

2.0 Introdução:

Este capítulo começa com uma descrição do problema geral de controle ótimo. A seguir, na Seção 2.2, trata-se o caso quadrático linear determinístico discreto (QLDD), que é o assunto deste trabalho. Expressa-se o problema matemáticamente, e explica-se as simplificações feitas para a sua programação.

Um resumo dos conceitos da programação dinâmica segue, com um exemplo numérico. O algoritmo para a solução do problema QLDD (assunto do Capítulo 3) é fundamentado neste método. Na Seção 2.4, trata-se o caso especial do problema de "tracking", e na Seção 2.5 é dada a abordagem recomendada para vetores defasados de estado e controle. Finalmente, na Seção 2.6 deriva-se a "forma final" das equações de sistema, e explica-se os vários tipos de multiplicadores.

2.1 O problema de controle ótimo:

No problema de controle ótimo tenta-se minimizar uma função objetivo através do tempo. Isto é feito através da seleção de vetores de controle, que influenciam os valores subsequentes dos vetores de estado. Supõe-se que existe pleno poder de seleção sobre os vetores de controle, mas valores mais elevados terão penalidades (custos) maiores na função objetivo.

Os vetores de estado também são influenciados pelos vetores exógenos. Os valores dos vetores exógenos são determinados exteriormente, por forças externas. Não podem ser controlados, nem influenciados. Porém, são tomados como conhecidos.

Finalmente, o vetor de estado é influenciado por seu próprio valor no período anterior. Matemáticamente, para o intervalo de tempo k, tem-se:

(1) $x_{k+1} = f(x_k, u_k, z_k)$

onde x_k é o vetor de estado, u_k é o vetor de controle, e z_k é o vetor exógeno do período k. No problema geral de controle ótimo, $f(x_k, u_k, z_k)$ pode ser linear ou não linear.

Em uma aplicação macro-econômica o nível de desemprego poderia ser um elemento do vetor de estado, o orçamento governamental um elemento do vetor de controle, e o preço de combustível importado parte do vetor exógeno. O orçamento governamental seria manipulado para tentar reduzir o desemprego. Tal manipulação teria que ser feito dentro dos limites impostos pelo preço de combustível importado.

O problema pode ser tanto estocástico quanto determinístico. No caso estocástico pode existir incerteza sobre o estado atual do sistema, incerteza com respeito da resposta do sistema às seleções do vetor de controle, e incerteza com respeito aos eventos futuros. No problema determinístico, supõe-se que tais incertezas não existem.

O problema também pode ser classificado como sendo em tempo contínuo ou discreto. No caso contínuo, a função objetivo é uma integral sobre o tempo, e as equações de sistema são equações lineares diferenciais. No caso discreto, a função objetivo é uma soma sobre o tempo, e as equações de sistema são equações lineares de diferenças. O problema discreto em geral é mais simples, e a sua forma é mais conveniente para resolução usando computadores.

2.2 O problema quadrático linear determinístico discreto:

O problema de controle ótimo que trata-se aqui é o problema quadrático linear determinístico discreto. O problema é assim chamado porque se quer otimizar uma função quadrática (a função objetivo) sujeita a um conjunto de equações lineares. Não existe incerteza, e o tempo é considerado em intervalos discretos. A forma padrão é:

Achar u_k para $k = 0, 1, \ldots, N-1$

(2) $$\text{Min } J = \tfrac{1}{2}x_N' W_N x_N + w_N' x_N + \sum_{k=0}^{N-1} \left(\tfrac{1}{2}x_k' W_k x_k + w_k' x_k + \tfrac{1}{2}u_k' L_k u_k + l_k' u_k\right)$$

Sujeito a:

(3) $$x_{k+1} = A_k x_k + B_k u_k + C_k + D_k z_k$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

Onde:

x_k é um vetor de estado de ST elementos.

u_k é um vetor de controle de CON elementos.

z_k é um vetor exógeno de EX elementos.

W_k é uma matriz de penalidades (ST x ST).

w_k é um vetor de penalidades de ST elementos.

L_K é uma matriz de penalidades (CON x CON).

l_k é um vetor de penalidades de CON elementos.

A_k é uma matriz (ST x ST).

B_k é uma matriz (ST x CON).

C_k é um vetor (ST x 1).

D_k é uma matriz (ST x EX).

N é o número de períodos no problema (horizonte).

Então, o problema consiste em escolher os vetores de controle u_k do período inicial até o penúltimo período de tal maneira que a função objetivo (2) seja minimizada. Tal seleção influencia os vetores de estado x_k. Os dois tipos de vetor aparecem na função objetivo.

O problema que pode ser resolvido pelo PQL é uma simplificação deste. No programa, as matrizes de penalidade da função objetivo e as matrizes constantes das equações de sistema não variam com o tempo. Isto é:

$$(4)\quad A_0 = A_1 = A_2 = \dots = A_{N-1} = A$$

$$B_0 = B_1 = B_2 = \dots = B_{N-1} = B$$

$$C_0 = C_1 = C_2 = \dots = C_{N-1} = C$$

$$D_0 = D_1 = D_2 = \dots = D_{N-1} = D$$

Simplificando ainda mais tem-se:

$$(5)\quad W_0 = W_1 = W_2 = \dots = W_{N-1} = W$$

$$L_0 = L_1 = L_2 = \dots = L_{N-1} = L$$

$$w_0 = w_1 = w_2 = \dots = w_{N-1} = w$$

$$l_0 = l_1 = l_2 = \dots = l_{N-1} = l$$

Porém, $W_k \neq W_N$, e $w_k \neq w_N$.

As simplificações em W, L, w e l implicam penalidades iguais na função objetivo para todos os períodos, de T = 0 até T = N-1. A penalidade no último período pode ser diferente dos períodos anteriores nas matrizes de penalidade dos vetores de estado. Assim o programa permite W_k, k = 0,1,...,N-1 diferente de W_N e w_k diferente de w_N.

Observe-se que nesta apresentação os nomes das variáveis não são totalmente consistentes com aqueles do PQL. Turbo Pascal não faz uma distinção entre letras minúsculas e maiúsculas nos nomes das variáveis. Para não confundir w com W, no PQL, usa-se w_ para w. Semelhantemente, usa-se l_ para l.

A derivação do método de solução do problema QLDD pode ser feita ou através do uso de multiplicadores de Lagrange ou da programação dinâmica.

Logo a seguir são apresentadados os conceitos básicos da programação dinâmica, e no Capítulo 3 esta técnica é usada na derivação do algoritmo para o problema QLDD.

2.3 Programação dinâmica:

2.3.0 Introdução:

Esta seção introduz a técnica de programação dinâmica, usada no Capítulo 3 para desenvolver o método de solução para problemas quadráticos lineares de controle ótimo. Primeiro, é dada uma descrição geral da programação dinâmica. Na próxima seção apresenta-se um exemplo numérico de programação de produção, extraído de Hastings (1973, p. 45). A intenção deste exemplo é de introduzir a terminologia e concretizar as noções fundamentais. Em seguida, resume-se um exemplo de Kendrick (1981, p. 11), com o objetivo de introduzir os conceitos da regra de realimentação e as matrizes Ricatti.

2.3.1 Descrição Geral de programação dinâmica:

Em programação dinâmica considera-se um sistema que pode seguir vários possíveis caminhos. O caminho é influenciado por um decisor que, em cada estágio, escolhe uma de várias ações possíveis. O sistema gera, em função das ações escolhidas, uma série de retornos, que podem ser tempo, distância, ou dinheiro. O decisor quer então otimizar uma função destes retornos gerados pelo sistema. Um exemplo adaptado de Hastings ilustra estas idéias.

2.3.2 Exemplo de programação de produção:

O gerente de uma fábrica de barcos tem um horizonte de vendas de três mêses, após o qual a fábrica será fechada. Durante este intervalo ele quer minimizar os custos de "overhead" e de estoque. Os custos de "overhead" ($4000) são incorridos quando a fábrica produz um ou mais barcos. São zero quando a produção é zero. O custo de manter estoque é de $1000 por barco por mês. Além disso, a produção máxima é de quatro barcos por mês e o estoque máximo é de três barcos. As vendas contratadas aparecem no Quadro 1:

Mês	mar	abr	mai
Barcos	2	1	3

QUADRO 1 - Vendas de barcos

O estoque inicial é zero. Todos os pedidos têm que ser fornecidos.

Agora, é necessário definir alguns termos. Vamos chamar cada mês um estágio do problema, simbolicamente n. O estoque é o estado do sistema, referido como i no período n e j no período n+1. A decisão de produção é a ação, k. A demanda do período é d. O custo incorrido em um estágio é o retorno, r(i,k). E uma função do seu estoque inicial e da ação. O valor ótimo de um estado, f(n,i), é o menor custo do estado atual até o estado final.

Uma característica da programação dinâmica é a frequente resolução dos problemas de trás para frente. Neste espírito, considera-se o gerente como já estando no último período (estágio), maio. Ele quer satisfazer toda a demanda de maio e terminar o período com estoque (estado) zero. Então, se ele não tiver nenhum barco em estoque no início de maio, ele vai produzir (ação) três. Se ele tiver um barco no estoque, produzirá dois. Com estoque dois, produzirá um barco, e com estoque três, não produzirá nenhum. Resumindo, tem-se o Quadro 2.

Estoque	Produção	Custo de Estoque	Custo de "Overhead"	Custo Total
0	3	0	4	4
1	2	1	4	5
2	1	2	4	6
3	0	3	0	3

QUADRO 2 - Custos de Maio

Para cada mês, o nível de produção, junto com o estoque inicial e as vendas do mês determinam o estoque inicial no mês seguinte. Isto pode ser expresso através da equação:

(6) $\text{Estoque}_{n+1} = \text{Estoque}_n + \text{Producao}_n - \text{Demanda}_n$

Em símbolos, tem-se:

(7) $j = i + k - d$

Esta expressão é chamada a equação de transição.

Agora, o gerente está em abril, o penúltimo período. Baseado no seu estoque inicial e os custos futuros de maio, ele pode determinar uma ação. Ele deve estar consciente das restrições de produção e estoque. Suponha que o estoque no início de abril seja zero. Neste caso ele tem que fabricar pelo menos um barco para satisfazer a demanda de maio. Poderia produzir até quatro.

No caso do gerente produzir apenas um barco, ele incorreria em abril o custo de "overhead", $4,000. Além disso, ele iniciaria maio com estoque zero, que implicaria um custo futuro de $4,000. Então o custo nos últimos dois períodos seria $8,000, se ele começasse abril com estoque zero e fabricasse um barco.

Se ele fabricasse dois barcos, ele pagaria $4,000 em abril. Depóis de vender o barco contratado para este mês, ficaria com um barco no início de maio, o que implica um custo de $5,000. O custo total de fabricar dois barcos em abril, supondo que o estoque inicial foi zero, é de $9,000.

Fabricando três barcos, o custo seria $4,000 em abril e $6,000 em maio. O total seria $10,000.

Fabricando quatro barcos, o custo seria $4,000 em abril e $3,000 em maio. O custo total seria $7,000.

Se o estoque for zero no início de abril, a melhor alternativa é de fabricar quatro barcos no mês. Através da equação de transição sabe-se que isso implica um estoque de três no início de maio, e nenhum barco produzido em maio. Vê-se a melhor alternativa na aplicação da equação de recorrência:

$$(8) \quad f(\text{abril}, i=0) = \underset{k=1,2,3,4}{\text{Min}} \{ r(\text{abril}, k) + f(\text{maio}, j=i+k-d) \}$$

Em palavras, o custo total ótimo desde abril com estoque zero é igual ao mínimo das somas do custo associado com a ação em abril e o custo que tal ação implica a partir de maio.

A idéia fundamental da programação dinâmica está refletida no segundo termo da minimização, que reflete o Princípio de Otimalidade de Bellman. Chegando em um certo ponto, o caminho até o ponto final não depende em como chegou-se a este ponto. Começando abril com um determinado estoque, as ações ótimas para chegar ao final de maio com estoque zero independem de que aconteceu antes de abril.

O resumo dos custos para os vários níveis de estoque e de produção segue no Quadro 3. Para cada estoque inicial, identifica-se a melhor política de produção por um asterisco. Um estoque inicial em abril de um barco oferece duas ações igualmente boas.

Estoque abril	Produção	Custo abril	Estoque maio	Custo maio	Custo total
0	1	4	0	4	8
0	2	4	1	5	9
0	3	4	2	6	10
0	4	4	3	3	7*
1	0	1	0	4	8*
1	1	5	1	4	9
1	2	5	2	5	10
1	3	5	3	8	8*
2	0	2	1	5	7*
2	1	6	2	6	12
2	2	6	3	3	9
3	0	3	2	6	9
3	1	7	3	3	8*

QUADRO 3 - Custos de abril e maio

Das tabelas acima, descartando as linhas sem asterisco, pode-se resumir os resultados para o mês de abril no Quadro 4.

Estoque abril	Produção Otimo	Custo Total (abril + futuro)
0	4	7
1	3	8
2	0	7
3	1	8

QUADRO 4 - Produção ótima para os níveis de estoque de abril

Com esta informação, o gerente considera o período inicial. Dado um estoque inicial de zero e contratos para dois barcos em março, ele pode fabricar entre dois e quatro barcos. Neste problema, as três opções têm o mesmo custo para março, $4,000. Novamente fazendo uma tabela:

Estoque março	Produção	Custo março	Estoque abril	Custo futuro	Custo total
0	2	4	0	7	11*
0	3	4	1	8	12
0	4	4	2	7	11*

QUADRO 5 - Custos de planos de produção a partir de março

Para minimizar os seus custos de "overhead" e estoque, o gerente tem duas estratégias possíveis. A primeira é de produzir dois barcos em março, quatro em abril, e nenhum em maio. A segunda é de produzir quatro em março, nenhum em abril, e dois em maio.

2.3.3 Um exemplo em transportes:

Este exemplo é usado por Kendrick (1982, p. 10-12) para desenvolver o algoritmo de solução para problemas determinísticos quadráticos lineares de controle ótimo. As suas contribuições principais são as idéias da regra de realimentação e das matrizes Ricatti.

Um avião voa de Nova Iorque para Londres. O piloto tenta minimizar os seus custos de combustível pela seleção da melhor rota, que depende do vento e do tempo.

Considera dois dias em que as condições de viagem são diferentes na primeira metade da viagem, mas iguais na segunda metade. Dois vôos nos dias diferentes claramente não teriam as mesmas rotas na primeira parte da viagem. Mas, suponha que eles começem a segunda metade no mesmo ponto. O resto da viagem seguiria a mesma rota para os dois vôos? Sim. A rota ótima na segunda metade depende só do vento e do tempo lá, e não da rota seguida na primeira metade. De novo, aparece a idéia fundamental da programação dinâmica.

Suponha, agora, que se pode associar a todos os pontos da viagem um caminho de mínimo custo até Londres. Se a informação fosse tabelada, daria para determinar a rota ótima a partir de qualquer ponto. Seria um questão de consultar a tabela para a direção do leme. Esta idéia pode ser usada para visualizar a regra de realimentação da forma:

$$(9)\quad u_k = G_k x_k + g_k$$

onde:

x_k é o vetor de estado dando a posição do avião

u_k é o vetor de controle dando as direções do leme

G_k é uma matriz de coeficientes

g_k é um vetor de coeficientes

Esta regra determina as posições dos controles do avião u_k a partir da posição do avião x_k. Para aplicá-la, os valores de G_k e g_k teriam que ser determinados para toda parte da viagem k. Então, tendo o vetor de estado x_k e as matrizes G_k e g_k pode-se calcular o vetor de controle u_k. Este conceito ficará mais claro depóis de ler a derivação do algoritmo no Capítulo 3.

Finalmente, imagine uma matriz e um vetor que juntos podem ser usados para calcular o custo mínimo de viagem da posição x_k até Londres em termos de x_k. O custo mínimo $J^*(x_k)$ seria:

$$(10)\quad J^*(x_k) = \tfrac{1}{2}x_k' K_k x_k + p_k' x_k$$

A matriz K_k e o vetor p_k são denominados de matrizes Ricatti. A maneira em que elas são calculadas e o papel delas no cálculo das matrizes da regra de realimentação, G_k e g_k, são assuntos do próximo capítulo.

Para terminar esta seção, deve-se notar que na maioria das aplicações, o k não representa a fase de uma viagem e sim um intervalo de tempo.

2.4 O problema de "tracking":

O problema QLDD de "tracking" é da forma:

Achar u_k $k = 0, 1, \ldots, N-1$ para

$$(11)\quad \text{Min } J = \tfrac{1}{2}[x_N - \bar{x}_N]'\bar{W}_N[x_N - \bar{x}_N] + \tfrac{1}{2}\sum_{k=0}^{N-1}\left([x_k - \bar{x}_k]'\bar{W}_k[x_k - \bar{x}_k] + [u_k - \bar{u}_k]'\bar{L}_k[u_k - \bar{u}_k]\right)$$

Sujeito a:

$$(12)\quad x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

$\bar{x}_k$ dado para $k = 0, 1, \ldots, N$

$\bar{u}_k$ dado para $k = 0, 1, \ldots, N-1$

Onde:

$\bar{x}_k$ é um vetor de estado desejado para período k.

$\bar{u}_k$ é um vetor de controle desejado para período k.

$\bar{W}_k$ é uma matriz de penalidades das variações dos vetores de estado dos seus valores desejados.

$\bar{L}_k$ é uma matriz de penalidades das variações dos vetores de controle dos seus valores desejados.

Normalmente as matrizes $\bar{W}$ e $\bar{L}$ são simétricas. Então, $\bar{W}' = \bar{W}$.

A equivalência do problema de "tracking" com o problema original pode ser comprovada em três passos. Primeiro, multiplica-se as matrizes da função objetivo (11). O resultado aparece em (13). Segundo, descarta-se os termos constantes. A solução do problema, ou seja, os valores de u_k, não depende das constantes. Isto resulta em (14).

$$(13)\quad \text{Min } J = \tfrac{1}{2}x'_N \bar{W}_N x_N - \tfrac{1}{2}x'_N \bar{W}_N \bar{x}_N - \tfrac{1}{2}\bar{x}'_N \bar{W}_N x_N + \tfrac{1}{2}\bar{x}'_N \bar{W}_N \bar{x}_N + \tfrac{1}{2}\sum_{k=0}^{N-1} (x'_k \bar{W}_k x_k - x'_k \bar{W}_k \bar{x}_k - \bar{x}'_k \bar{W}_k x_k + \bar{x}'_k \bar{W}_k \bar{x}_k$$

$$+ u'_k \bar{W}_k u_k - u'_k \bar{W}_k \bar{u}_k - \bar{u}'_k \bar{W}_k u_k + \bar{u}'_k \bar{W}_k \bar{u}_k)$$

$$(14)\quad \text{Min } J = \tfrac{1}{2}x'_N \bar{W}_N x_N - \tfrac{1}{2}x'_N \bar{W}_N \bar{x}_N - \tfrac{1}{2}\bar{x}'_N \bar{W}_N x_N + \tfrac{1}{2}\sum_{k=0}^{N-1} (x'_k \bar{W}_k x_k - x'_k \bar{W}_k \bar{x}_k - \bar{x}'_k \bar{W}_k x_k$$

$$+ u'_k \bar{L}_k u_k - u'_k \bar{L}_k \bar{u}_k - \bar{u}'_k \bar{L}_k u_k)$$

Terceiro, efetua-se as substituições do Quadro 6, notando as igualdades em (15). Ao multiplicar o interior da soma por meio tem-se (16), que é a forma geral da função objetivo.

Geral	"Tracking"
W_k	$\bar{W}_k$
w_k	$-\bar{W}_k \bar{x}_k$
W_k	$\bar{W}_k$
w_k	$-\bar{W}_k \bar{x}_k$
L_k	$\bar{L}_k$
l_k	$-\bar{L}_k \bar{u}_k$

Quadro 6 - Equivalência do problema de "tracking"

$$(15)\quad -x'_k \bar{W}_k \bar{x}_k = x'_k w_k = w'_k x_k$$

$$-\bar{x}'_k \bar{W}_k x_k = -x'_k \bar{W}_k \bar{x}_k = x'_k w_k = w'_k x_k$$

$$(16)\quad \text{Min } J = \tfrac{1}{2}x'_N W_N x_N + w'_N x_N + \sum_{k=0}^{N-1} (\tfrac{1}{2}x'_k W_k x_k + w'_k x_k + \tfrac{1}{2}u'_k L_k u_k + l'_k u_k)$$

No caso geral da Seção 2.2, usava-se um vetor constante de valores para l_k. O w_k foi um vetor constante para os períodos 1 até N-1, e um outro para período N. Isto simplifica a formulação do problema e entrada dos seus dados. Porém, no caso de "tracking", as equivalências do Quadro 6 são usadas para calcular vetores diferentes em períodos diferentes, para w_k e l_k.

Finalmente, nota-se que o valor da função objetivo (11) será, em geral, diferente do que aquele de (16). Porém, a solução de um problema com função objetivo (11) é a mesma que um problema com função objetivo (16).

2.5 Vetores defasados:

No modelo 2.2, tem-se a seguinte equação de sistema:

$$(3) \quad x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

$$\text{para } k = 0, 1, \ldots, N-1$$

Esta equação é chamada uma equação de diferença da primeira ordem porque, na determinação de x_{k+1}, apenas usam-se vetores com subscrito k. As vezes, porém, os vetores têm um efeito defasado, e x_{k+1} é uma função de vetores de períodos antes do período k. Se for preciso usar até x_{k-m}, a equação seria da ordem m. Em geral, uma diferença de m nos subscritos da equação implica uma equação de diferença da ordem m+1.

Considere x_{k+1} que é uma função não apenas do vetores do período anterior k, mas também do período k-1. Seria, então, uma equação de segunda ordem. Tem-se:

$$(17) \quad x_{k+1} = A_0 x_k + A_1 x_{k+1} + B_0 u_k + B_1 u_{k-1} + C + D_0 z_k + D_1 z_{k-1}$$

$$\text{para } k = 0, 1, \ldots, N-1$$

Um problema com esta equação de sistema pode ser resolvida usando o método do Capitulo 3, mas primeiro tem que ser re-escrito na forma de uma equação de diferença de primeira ordem de mais variáveis. Usando a abordagem de Kendrick (1982, p. 8) definem-se três novos vetores:

(18) $$y_k = x_{k-1}$$

$$v_k = u_{k-1}$$

$$q_k = z_{k-1}$$

Agora, equação (18) pode ser re-escrita:

(19) $$x_{k+1} = A_0 x_k + A_1 y_k + B_0 u_k + B_1 v_k + C + D_0 z_k + D_1 q_k$$

para $k = 0, 1, \ldots, N-1$

O vetor de estado aumentado é definido como:

(20) $$s_k = \begin{bmatrix} x \\ y \\ v \\ q \end{bmatrix}_k$$

As equações (18) podem ser re-escritas:

(21) $$y_{k+1} = x_k$$

$$v_{k+1} = u_k$$

$$q_{k+1} = z_k$$

Então (19) pode ser escrita na forma:

$$(22)\quad \begin{bmatrix} x \\ y \\ v \\ q \end{bmatrix}_{k+1} = \begin{bmatrix} A_0 & A_1 & B_1 & D_1 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ v \\ q \end{bmatrix}_{k} + \begin{bmatrix} B_0 \\ 0 \\ I \\ 0 \end{bmatrix} u_k + \begin{bmatrix} D_0 \\ 0 \\ 0 \\ I \end{bmatrix} z_k + \begin{bmatrix} C \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Ou então:

$$(23)\quad s_{k+1} = A^*s_k + B^*u_k + D^*z_k + C^*$$

Onde:

$$(24)\quad A^* = \begin{bmatrix} A_0 & A_1 & B_1 & D_1 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B^* = \begin{bmatrix} B_0 \\ 0 \\ I \\ 0 \end{bmatrix} \quad D^* = \begin{bmatrix} D_0 \\ 0 \\ 0 \\ I \end{bmatrix} \quad C^* = \begin{bmatrix} C \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

O procedimento para equações de diferença de maior ordem é comparável. Claramente, a criação de novas variáveis que resultam deste método aumenta o tamanho das matrizes construídas na resolução do problema.

2.6 Multiplicadores:

Agora, trata-se os multiplicadores, que podem ser usados para ver a influência de um elemento sobre um outro, considerando o tempo. Primeiro, uma derivação da "forma geral" das equações de sistema será dada. Depóis, os quatro tipos de multiplicadores descritos por Chow (1975, p. 107) serão definidos.

A seguir são as equações de sistema para os períodos k+1, k, k-1, e k-2:

(25) $x_{k+1} = Ax_k + Bu_k + C + Dz_k$

(26) $x_k = Ax_{k-1} + Bu_{k-1} + C + Dz_{k-1}$

(27) $x_{k-1} = Ax_{k-2} + Bu_{k-2} + C + Dz_{k-2}$

(28) $x_{k-2} = Ax_{k-3} + Bu_{k-3} + C + Dz_{k-3}$

O x_k da equação (25) pode ser substituído pela equação (26).

(29) $x_{k+1} = A\,(Ax_{k-1} + Bu_{k-1} + C + Dz_{k-1}) + Bu_k + C + Dz_k$

$x_{k+1} = A^2x_{k-1} + ABu_{k-1} + AC + ADz_{k-1} + Bu_k + C + Dz_k$

O x_{k-1} da equação (29) pode ser substituído pela equação (27).

(30) $x_{k+1} = A^2\,(Ax_{k-2} + Bu_{k-2} + C + Dz_{k-2}) + ABu_{k-1} + AC + ADz_{k-1} + Bu_k + C + Dz_k$

$x_{k+1} = A^3x_{k-2} + A^2Bu_{k-2} + A^2C + A^2Dz_{k-1} + ABu_{k-1} + AC + ADz_{k-1} + Bu_k + C + Dz_k$

Da mesma maneira, o x_{k-2} da equação (30) pode ser substituído pela equação (28).

(31) $x_{k+1} = A^3\,(Ax_{k-3} + Bu_{k-3} + C + Dz_{k-3})$

$+ A^2Bu_{k-2} + A^2C + A^2Dz_{k-1} + ABu_{k-1} + AC + ADz_{k-1} + Bu_k + C + Dz_k$

$x_{k+1} = A^4x_{k-3} + A^3Bu_{k-3} + A^3C + A^3Dz_{k-3}$

$+ A^2Bu_{k-2} + A^2C + A^2Dz_{k-1} + ABu_{k-1} + AC + ADz_{k-1} + Bu_k + C + Dz_k$

Continuando a substituição, finalmente chega-se à forma final:

$$(32) \quad x_{k+1} = A^{k+1}x_0 + Bu_k + ABu_{k-1} + A^2Bu_{k-2} + \ldots + A^kBu_1$$

$$+ Dz_k + ADz_{k-1} + A^2Dz_{k-2} + \ldots + A^kDz_1 + C + AC + A^2C + A^3C + \ldots + A^kC$$

Esta forma será usada para definir os multiplicadores citados por Chow.

O multiplicador de impacto do j-ésimo elemento de controle no i-ésimo elemento de estado é dada pelo i-j-ésimo elemento da matrix B.

Os multiplicadores defasados são as derivadas dos elementos de estado com respeito aos elementos defasados de controle. De (32) o multiplicador defasado do j-ésimo elemento de controle defasado k períodos no i-ésimo elemento de estado é o i-j-ésimo elemento da matrix A^kB.

Os multiplicadorés de prazo intermediário medem os efeitos somados de mudanças nos elementos de controle que persistiram durante vários períodos. O multiplicador de prazo intermediário do j-ésimo elemento de controle que persiste durante m períodos no i-ésimo elemento de estado é o i-j-ésimo elemento da soma de matrizes $B + AB + A^2B + \ldots + A^{m-1}B$.

Se o multiplicador de prazo intermediário cresce rapidamente, pode indicar instabilidade no sistema. Neste caso, o multiplicador de muito longo prazo não existiria.

O multiplicador de muito longo prazo existe se todas as raizes de polinômio características de A tiveram valor absoluto menor que 1. Neste caso, é o limite dos multiplicadores de prazo intermediário com m chegando ao infinito. Segundo Chow, se (I-A) for uma matriz não singular, a matriz de multiplicadores de prazo intermediário pode ser escrita:

$$(33) \quad (I + A + A^2 + \ldots + A^{m-1})B = (I - A)^{-1}(I-A^m)B$$

No limite, A^m tende a 0 e a matriz dos multiplicadores de muito longo prazo pode ser escrita: $(I - A)^{-1}\,B$.

CAPÍTULO 3: DERIVAÇÃO DO ALGORITMO

3.0 Introdução:

Neste capítulo é apresentada a derivação do algoritmo para resolução de problemas do tipo quadrático linear determinístico discreto. Examina-se dois casos muito semelhantes. No primeiro caso, o vetor de controle tem um efeito defasado: os efeitos de mexer no vetor de controle em período t só serão sentidos no período t+1. No segundo caso os efeitos são imediatos.

3.1 Vetor de controle com impacto defasado:

A expressão matemática do problema é repetida a seguir.

Achar u_k para $k = 0, 1, \ldots, N-1$

$$(2)\quad \text{Min } J = \tfrac{1}{2}x_N'W_Nx_N + w_N'x_N + \sum_{k=0}^{N-1}\left(\tfrac{1}{2}x_k'W_kx_k + w_k'x_k + \tfrac{1}{2}u_k'L_ku_k + l_k'u_k\right)$$

Sujeito a:

$$(3)\quad x_{k+1} = A_kx_k + B_ku_k + C_k + D_kz_k$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

Aplicando o método de programação dinâmica na derivação do método de solução, deve-se começar no último período e progredir de trás para frente. Da função objetivo (2), sabe-se que o custo no último período será:

$$(34)\quad J^*(x_N) = J^*(N) = \tfrac{1}{2}x_N'W_Nx_N + w_N'x_N$$

No período final N as matrizes de penalidade da função objetivo são iguais às matrizes Ricatti.

$$(35)\quad K_N = W_N$$

$$p_N = w_N$$

Agora é necessario calcular os custos do período penúltimo até o período final, que seria a soma dos custos dos dois períodos. O custo do penúltimo período N-1 pode ser expresso usando a função $L_{N-1}(x_{N-1},u_{N-1})$. Da função objetivo sabe-se que:

$$(36)\quad L_{N-1}(x_{N-1},u_{N-1}) = \tfrac{1}{2}x'_{N-1}W_{N-1}x_{N-1} + w'_{N-1}x_{N-1} + \tfrac{1}{2}u'_{N-1}\Lambda u_{N-1} + l'_{N-1}u_{N-1}$$

Então, fazendo a soma de (34) e (36) tem-se:

$$(37)\quad J*(N-1) = \min_{U_{N-1}} \{ J*(N) + L_{N-1}(x_{N-1},u_{N-1}) \}$$

Substituindo (34) e (36) em (37) tem-se:

$$(38)\quad J*(N-1) = \min_{U_{N-1}} \{ \tfrac{1}{2}x'_N W_N x_N + w'_N x_N + \tfrac{1}{2}x'_{N-1}W_{N-1}x_{N-1} + w'_{N-1}x_{N-1} + \tfrac{1}{2}u'_{N-1}\Lambda u_{N-1} + l'_{N-1}u_{N-1} \}$$

Pode-se eliminar x_N de (38) pela substituição das equações de sistema (3). Substituindo:

$$(3)\quad x_K = Ax_{K-1} + Bu_{K-1} + C + Dz_{K-1}$$

resulta em:

$$\begin{aligned}(39)\quad J*(N-1) = \min_{U_{N-1}} \{ & \tfrac{1}{2}(x'_{N-1}A' + u'_{N-1}B' + C' + z'_{N-1}D')K_N(Ax_{N-1} + Bu_{N-1} + C + Dz_{N-1}) \\ & + p'_N(Ax_{N-1} + Bu_{N-1} + C + Dz_{N-1}) + \tfrac{1}{2}x'_{N-1}W_{N-1}x_{N-1} \\ & + w'_{N-1}x_{N-1} + \tfrac{1}{2}u'_{N-1}\Lambda u_{N-1} + l'_{N-1}u_{N-1} \}\end{aligned}$$

Multiplicando, tem-se (40). Assim, J*(N-1) fica expressa em termos de x_{N-1} e u_{N-1}.

$$(40)\quad J^*(N-1) = \min_{u_{N-1}} \{ \tfrac{1}{2}x'_{N-1} A'K_N A x_{N-1} + \tfrac{1}{2}x'_{N-1} A'K_N B u_{N-1} + \tfrac{1}{2}x'_{N-1} A'K_N C + \tfrac{1}{2}x'_{N-1} A'K_N D z_{N-1}$$

$$+ \tfrac{1}{2}u'_{N-1} B'K_N A x_{N-1} + \tfrac{1}{2}u'_{N-1} B'K_N B u_{N-1} + \tfrac{1}{2}u'_{N-1} B'K_N C + \tfrac{1}{2}u'_{N-1} B'K_N D z_{N-1}$$

$$+ \tfrac{1}{2}C'K_N A x_{N-1} + \tfrac{1}{2}C'K_N B u_{N-1} + \tfrac{1}{2}C'K_N C + \tfrac{1}{2}C'K_N D z_{N-1}$$

$$+ \tfrac{1}{2}z'_{N-1} D'K_N A x_{N-1} + \tfrac{1}{2}z'_{N-1} D'K_N B u_{N-1} + \tfrac{1}{2}z'_{N-1} D'K_N C + \tfrac{1}{2}z'_{N-1} D'K_N D z_{N-1}$$

$$+ p'_N A x_{N-1} + p'_N B u_{N-1} + p'_N C + p'_N D z_{N-1} + \tfrac{1}{2}x'_{N-1} W_{N-1} x_{N-1} + w'_{N-1} x_{N-1}$$

$$+ \tfrac{1}{2}u'_{N-1} L u_{N-1} + l'_{N-1} u_{N-1} \}$$

Agrupando termos, a expressão pode ser escrita como em (41):

$$(41)\quad J^*(N-1) = \min_{u_{N-1}} \{ \tfrac{1}{2}x'_{N-1} PHI_{N-1} x_{N-1} + \tfrac{1}{2}u'_{N-1} THETA_{N-1} u_{N-1} + x'_{N-1} PSI_{N-1} u_{N-1} + phi'_{N-1} x_{N-1}$$

$$+ theta'_{N-1} u_{N-1} + mu_{N-1} \}$$

Onde:

$$PHI_{N-1} = A'K_N A + W_{N-1}$$

$$THETA_{N-1} = B'K_N B + L_{N-1}$$

$$PSI_{N-1} = A'K_N B$$

$$phi_{N-1} = A'K_N C + A'K_N D z_{N-1} + A'p_N + w_{N-1}$$

$$theta_{N-1} = B'K_N C + B'K_N D z_{N-1} + B'p_N + l_{N-1}$$

$$mu_{N-1} = \tfrac{1}{2}C'K_N C + z'_{N-1} D'K_N C + \tfrac{1}{2}z'_{N-1} D'K_N D z_{N-1} + p'_N C + p'_N D z_{N-1}$$

Minimizando com respeito de u_{N-1}, toma-se a derivada. O resultado fica:

$$(42)\quad u'_{N-1} THETA_{N-1} + x'_{N-1} PSI_{N-1} + theta'_{N-1} = 0$$

$$THETA'_{N-1} u_{N-1} + PSI'_{N-1} x_{N-1} + theta_{N-1} = 0$$

$$u_{N-1} + (THETA'_{N-1})^{-1} PSI'_{N-1} x_{N-1} + (THETA'_{N-1})^{-1} theta_{N-1} = 0$$

$$u_{N-1} = -(THETA'_{N-1})^{-1} PSI'_{N-1} x_{N-1} - (THETA'_{N-1})^{-1} theta_{N-1}$$

Simplificando, (42) pode ser expressa na forma de (43).

$$(43)\qquad u_{N-1} = G_{N-1} x_{N-1} + g_{N-1}$$

Onde:

$$G_{N-1} = -(THETA'_{N-1})^{-1} PSI'_{N-1}$$

$$g_{N-1} = -(THETA'_{N-1})^{-1} theta_{N-1}$$

Esta forma é a da regra de realimentação descrita na Seção 2.3.3. Porém, constitue só a regra para o período N-1. Deseja-se a regra de realimentação para o período geral, k.

Para fazer isso, repete-se o procedimento feito acima. Primeiro, em (44) expressa-se o custo J*(N-2) como a soma de J*(N-1) e $L_{N-2}(x_{N-2},u_{N-2})$. O segundo termo desta expressão $L_{N-2}(x_{N-2},u_{N-2})$ pode ser facilmente obtido da função objetivo (2). Aparece em (45).

$$(44)\qquad J^*(N-2) = \min_{u_{N-2}} \{ J^*(N-1) + L_{N-2}(x_{N-2},u_{N-2}) \}$$

$$(45)\qquad L_{N-2}(x_{N-2},u_{N-2}) = \tfrac{1}{2}x'_{N-2} W_{N-2} x_{N-2} + w'_{N-2} x_{N-2} + \tfrac{1}{2}u'_{N-2} L_{N-2} u_{N-2} + l'_{N-2} u_{N-2}$$

J*(N-1), o primeiro termo de (44), é mais problemático. A forma de (41) não é conveniente. Então, substituindo a regra de realimentação (43) em (41) pode-se eliminar o termo u_{N-1}. A substituição aparece em (46) é o agrupamento de termos resulta em (47).

$$(46)\qquad J^*(N-1) = \min_{u_{N-1}} \{ \tfrac{1}{2}x'_{N-1} PHI_{N-1} x_{N-1} + \tfrac{1}{2}(x'_{N-1} G'_{N-1} + g'_{N-1}) THETA_{N-1} (G_{N-1} x_{N-1} + g_{N-1})$$

$$+ x'_{N-1} PSI_{N-1} (G_{N-1} x_{N-1} + g_{N-1}) + phi'_{N-1} x_{N-1} + theta'_{N-1} (G_{N-1} x_{N-1} + g_{N-1}) + mu_{N-1} \}$$

$$= \min_{u_{N-1}} \{ \tfrac{1}{2}x'_{N-1} PHI_{N-1} x_{N-1} + \tfrac{1}{2}x'_{N-1} G'_{N-1} THETA_{N-1} G_{N-1} x_{N-1} + \tfrac{1}{2}x'_{N-1} G'_{N-1} THETA_{N-1} g_{N-1}$$

$$+ \tfrac{1}{2}g'_{N-1} THETA_{N-1} G_{N-1} x_{N-1} + \tfrac{1}{2}g'_{N-1} THETA_{N-} g_{N-1} + x'_{N-1} PSI_{N-1} G_{N-1} x_{N-1}$$

$$+ x'_{N-1} PSI_{N-1} g_{N-1} + phi'_{N-1} x_{N-1} + theta'_{N-1} G_{N-1} x_{N-1} + theta'_{N-1} g_{N-1} + mu_{N-1} \}$$

$$(47) \quad J^*(N-1) = \tfrac{1}{2}x'_{N-1}K_{N-1}x_{N-1} + p'_{N-1}x_{N-1}$$

$$K_{N-1} = PHI_{N-1} + G'_{N-1}\,THETA_{N-1}\,G_{N-1} + 2\cdot PSI_{N-1}\,G_{N-1}$$

$$p_{N-1} = G'_{N-1}\,THETA_{N-1}\,g_{N-1} + PSI_{N-1}\,g_{N-1} + phi'_{N-1} + G'_{N-1}\,theta_{N-1}$$

Em (47), os custos dos últimos dois períodos aparecem somente como uma função de x_{N-1}. Lembrando o comentário da Seção 2.3.3, as matrizes K_{N-1} e p_{N-1} de (47) são as matrizes Ricatti para o período N-1.

Substituindo (45) e (47) em (44) resulta em (48).

$$(48) \quad J^*(N-2) = \tfrac{1}{2}x'_{N-1}K_{N-1}x_{N-1} + p'_{N-1}x_{N-1}$$

$$+ \tfrac{1}{2}x'_{N-2}W_{N-2}x_{N-2} + w'_{N-2}x_{N-2} + \tfrac{1}{2}u'_{N-2}L_{N-2}u_{N-2} + l'_{N-2}u_{N-2}$$

Para eliminar x_{N-1} aplica-se as equações de sistema para obter (49). A multiplicação e o agrupamento de termos resulta em (50).

$$(49) \quad J^*(N-2) = \tfrac{1}{2}(x'_{N-2}A' + u'_{N-2}B' + C' + z'_{N-2}B')K_{N-1}(Ax_{N-2} + Bu_{N-2} + C + Dz_{N-2})$$

$$+ p'_{N-1}(Ax_{N-2} + Bu_{N-2} + C + Dz_{N-2})$$

$$+ \tfrac{1}{2}x'_{N-2}W_{N-2}x_{N-2} + w'_{N-2}x_{N-2} + \tfrac{1}{2}u'_{N-2}L_{N-2}u_{N-2} + l'_{N-2}u_{N-2}$$

$$J^*(N-2) = \min_{u_{N-2}} \{ \tfrac{1}{2}x'_{N-2}A'K_{N-1}Ax_{N-2} + \tfrac{1}{2}x'_{N-2}A'K_{N-1}Bu_{N-2} + \tfrac{1}{2}x'_{N-2}A'K_{N-1}C + \tfrac{1}{2}x'_{N-2}A'K_{N-1}Dz_{N-2}$$

$$+ \tfrac{1}{2}u'_{N-2}B'K_{N-1}Ax_{N-2} + \tfrac{1}{2}u'_{N-2}B'K_{N-1}Bu_{N-2} + \tfrac{1}{2}u'_{N-2}B'K_{N-1}C + \tfrac{1}{2}u'_{N-2}B'K_{N-1}Dz_{N-2}$$

$$+ \tfrac{1}{2}C'K_{N-1}Ax_{N-2} + \tfrac{1}{2}C'K_{N-1}Bu_{N-2} + \tfrac{1}{2}C'K_{N-1}C + \tfrac{1}{2}C'Dz_{N-2}$$

$$+ \tfrac{1}{2}z'_{N-2}D'K_{N-1}Ax_{N-2} + \tfrac{1}{2}z'_{N-2}D'K_{N-1}Bu_{N-2} + \tfrac{1}{2}z'_{N-2}D'K_{N-1}C + \tfrac{1}{2}z'_{N-2}D'K_{N-1}Dz_{N-2}$$

$$+ p'_N Ax_{N-2} + p'_N Bu_{N-2} + p'_N C + p'_N Dz_{N-2} + \tfrac{1}{2}x'_{N-2}W_{N-2}x_{N-2} + w'_{N-2}x_{N-2}$$

$$+ \tfrac{1}{2}u'_{N-2}Lu_{N-2} + l'_{N-2}u_{N-2} \}$$

(50) $$J^*(N-2) = \min_{u_{N-2}} \{ \tfrac{1}{2}x'_{N-2}\,PSI_{N-2}\,x_{N-2} + \tfrac{1}{2}\,u'_{N-2}\,THETA_{N-2}\,u_{N-2} + x'_{N-2}\,PHI_{N-2}\,u_{N-2} + psi'_{N-2}\,x_{N-2} + theta'_{N-2}\,u_{N-2} + mu_{N-2} \}$$

Onde:

$$PHI_{N-2} = A'K_{N-1}A + W_{N-2}$$

$$THETA_{N-2} = B'K_{N-1}B + L_{N-2}$$

$$PSI_{N-2} = A'K_{N-2}B$$

$$phi_{N-2} = A'K_{N-1}C + A'K_{N-1}Dz_{N-2} + A'p_{N-2} + w_{N-2}$$

$$theta_{N-2} = B'K_{N-1}C + B'K_{N-1}Dz_{N-2} + B'p_{N-1} + l_{N-2}$$

$$mu_{N-2} = \tfrac{1}{2}C'K_{N-1}C + z'_{N-2}D'K_{N-1}C + \tfrac{1}{2}z'_{N-2}B'K_{N-1}Dz_{N-2} + p'_{N-1}C + p'_{N-1}Dz_{N-2}$$

Como foi feito anteriormente, toma-se a derivada de (50) com respeito de u_{N-2} em (51). Colocando u_{N-2} em evidência resulta em (52).

(51) $$u'_{N-2}\,THETA_{N-2} + x'_{N-2}\,PSI_{N-2} + theta'_{N-2} = 0$$

$$THETA'_{N-2}\,u_{N-2} + PSI'_{N-2}\,x_{N-2} + theta_{N-2} = 0$$

$$u_{N-2} + (THETA'_{N-2})^{-1}PSI'_{N-2}\,x_{N-2} + (THETA'_{N-2})^{-1}theta_{N-2} = 0$$

$$u_{N-2} = -(THETA'_{N-2})^{-1}PSI'_{N-2}\,x_{N-2} - (THETA'_{N-2})^{-1}theta_{N-2}$$

(52) $$u_{N-2} = G_{N-2}\,x_{N-2} + g_{N-2}$$

Onde:

$$G_{N-2} = -(THETA'_{N-2})^{-1}PSI'_{N-2}$$

$$g_{N-2} = -(THETA'_{N-2})^{-1}theta_{N-2}$$

A equação (52) é a regra de realimentação para N-2. Nota-se que (52) tem a mesma forma que a regra de realimentação para N-1 em (43). Seguindo, pode-se eliminar u_{N-2} de (50), substituindo (52). Esta operação aparece em (53). Agrupando termos, tem-se (54).

$$(53)\quad J^*(N-2) = \min_{u_{N-2}} \{ \tfrac{1}{2}x'_{N-2} PHI_{N-2} x_{N-2} + \tfrac{1}{2}(x'_{N-2} G'_{N-2} + g'_{N-2}) THETA_{N-2} (G_{N-2} x_{N-2} + g_{N-2})$$
$$+ x'_{N-2} PSI_{N-2} (G_{N-2} x_{N-2} + g_{N-2}) + phi'_{N-2} x_{N-2} + theta'_{N-2} (G_{N-2} x_{N-2} + g_{N-2})$$
$$+ mu_{N-2} \}$$

$$= \min_{u_{N-2}} \{ \tfrac{1}{2}x'_{N-2} PHI_{N-2} x_{N-2} + \tfrac{1}{2}x'_{N-2} G'_{N-2} THETA_{N-2} G_{N-2} x_{N-2} + \tfrac{1}{2}x'_{N-2} G'_{N-2} THETA_{N-2} g_{N-2}$$
$$+ \tfrac{1}{2}g'_{N-2} THETA_{N-2} G_{N-2} x_{N-2} + \tfrac{1}{2}g'_{N-2} THETA_{N-2} g_{N-2} + x'_{N-2} PSI_{N-2} G_{N-2} x_{N-2}$$
$$+ x'_{N-2} PSI_{N-2} g_{N-2} + phi'_{N-2} x_{N-2} + theta'_{N-2} G_{N-2} x_{N-2} + theta'_{N-2} g_{N-2} + mu_{N-2} \}$$

$$(54)\quad J^*(N-2) = \tfrac{1}{2}x'_{N-2} K_{N-2} x_{N-2} + p'_{N-2} x_{N-2}$$

$$K_{N-2} = PHI_{N-2} + G'_{N-2} THETA_{N-2} G_{N-2} + 2 \cdot PSI_{N-2} G_{N-2}$$

$$p_{N-2} = G'_{N-2} THETA_{N-2} g_{N-2} + PSI_{N-2} g_{N-2} + phi_{N-2} + G'_{N-2} theta_{N-2}$$

A forma de (54) é a mesma de (47). Agora tem-se formas consistentes tanto para a regra de realimentação quanto para as matrizes Ricatti. A forma geral da regra de realimentação para o período k aparece em (55). A das matrizes Ricatti segue em (56), e as matrizes e vetores intermediários usados em tais equações aparecem em (57).

(55) $u_k = G_k x_k + g_k$

$G_k = -(\mathrm{THETA}_k)^{-1}\,\mathrm{PSI}_k$

$g_k = -(\mathrm{THETA}_k)^{-1}\,\mathrm{theta}_k$

(56) $K_k = \mathrm{PSI}_k + G_k\,\mathrm{THETA}_k\,G_k + 2\cdot\mathrm{PSI}_k G_k$

$p_k = G_k'\,\mathrm{THETA}_k g_k + \mathrm{PSI}_k g_k + \mathrm{phi}_k + G_k'\,\mathrm{theta}_k$

(57) $\mathrm{PHI}_k = A'K_{k+1}A + W_k$

$\mathrm{THETA}_k = B'K_{k+1}B + L_k$

$\mathrm{PSI}_k = A'K_{k+1}B$

$\mathrm{phi}_k = A'K_{k+1}C + A'K_{k+1}Dz_k + A'p_{k+1} + w_k$

$\mathrm{theta}_k = B'K_{k+1}C + B'K_{k+1}Dz_k + B'p_{k+1} + l_k$

$\mathrm{mu}_k = \frac{1}{2}C'K_{k+1}C + z_k'D'K_{k+1}C + \frac{1}{2}z_k'D'K_{k+1}Dz_k + p_{k+1}'C + p_{k+1}'Dz_k$

Agora, resume-se o método de solução do problema quadrático linear determinístico discreto de controle ótimo. Primeiro, alterna-se o cálculo das matrizes Ricatti K_k e p_k e as matrizes da regra de realimentação g_k e G_k. Isto é feito começando no penúltimo período N-1 e progredindo até o período inicial 0. Com g_k e G_k calculadas para todo k, pode-se aplicar a regra de realimentação e as equações de sistema. Com x_0 dado, aplica-se a regra de realimentação para determinar u_0. Então, com u_0 e x_0 usam-se as equações de sistema para determinar x_1. Este procedimento segue até que sejam calculadas x_k e u_k para todo k. Um resumo formal do algoritmo é apresentado no segundo Apêndice.

Algumas aplicações seguem no Capítulo 4.

3.2 Vetor de controle com impacto imediato:

A expressão matemática do problema com o vetor de controle com impacto imediato segue:

Achar u_k para $k = 1, \ldots, N$

(58) $\text{Min } J = \sum_{k=1}^{N} (\tfrac{1}{2} x_k' W_k x_k + w_k' x_k)$

Sujeito a:

(59) $x_{k+1} = Ax_k + Bu_{k+1} + C + Dz_{k+1}$

para $k = 1, \ldots, N-1$

com x_0 dado

Nota-se que neste caso, a função objetivo é uma soma de período um até N, e não de zero até N-1 como no caso anterior. Observa-se também que a função objetivo (58) só tem matrizes de penalidade para os vetores de estado, o que simplifica bastante a derivação do algoritmo. Para incluir o vetor de controle na função objetivo, deve-se embutí-lo no vetor de estado. Apresenta-se o procedimento na Seção 2.5, e um exemplo na Seção 4.1.

Nesta derivação, o vetor exógeno é também de impacto imediato. Lembra-se que o produto Dz_{k+1} é uma constante. Então, se o vetor exógeno fosse defasado o termo Dz_k substituiria Dz_{k+1} em todas as equações. Além disso, a derivação não mudaria.

Os passos da derivação para o caso atual seguem exatamente a mesma sequência que no caso do vetor de controle defasado. A regra de realimentação fica:

(60) $u_k = G_k x_{k-1} + g_k$

$G_k = -(\text{THETA}_k)^{-1}\,\text{PSI}_k$

$g_k = -(\text{THETA}_k)^{-1}\,\text{theta}_k$

As matrizes Ricatti ficam:

$$(61)\quad K_x = PHI_x + G_x THETA_x G_x + 2 \cdot PSI_x G_x$$

$$p_x = g_x' THETA_x G_x + g_x' PSI_x' + phi_x' + theta_x' G_x$$

Onde:

$$(62)\quad PHI_x = A'K_{x+1}A + W_x$$

$$THETA_x = B'K_{x+1}B$$

$$PSI_x = A'K_{x+1}B$$

$$phi_x = A'K_{x+1}C + A'K_{x+1}Dz_{x+1} + A'p_{x+1} + w_x$$

$$theta_x = B'K_{x+1}C + B'K_{x+1}Dz_{x+1} + B'p_{x+1}$$

$$mu_x = \tfrac{1}{2}C'K_{x+1}C + z_{x+1}'D'K_{x+1}C + \tfrac{1}{2}z_{x+1}'D'K_{x+1}Dz_{x+1} + p_{x+1}'C + p_{x+1}'Dz_{x+1}$$

Nota-se a semelhança com o caso anterior. A regra de realimentação só varia no subscrito do vetor de controle. Além disso, as matrizes intermediárias do primeiro caso (defasado) seriam iguais ao segundo (imediato) se as matrizes de penalidade para os vetores de controle do primeiro fossem zeradas.

Usando estas equações, o procedimento para a resolução do problema com vetor de controle imediato é o mesmo do caso defasado.

CAPÍTULO 4: APLICAÇÕES:

4.0 Introdução:

Este capítulo apresenta duas aplicações do algoritmo do capítulo anterior. Na Seção 4.1 trata-se um exemplo micro-econômico de planejamento de produção. Dados os custos relevantes e a demanda futura de um produto, determina-se níveis ótimos de produção e estoque, bem como o número de empregados e as horas extras trabalhadas por eles em cada período. Na Seção 4.2 apresenta-se um exemplo macro-econômico, em que tenta-se determinar as condições econômicas internacionais mais favoráveis para a redução da dívida externa e o crescimentó da economia brasileira.

4.1.1 Um problema de planejamento de produção: formulação:

Este exemplo é uma adaptação de um problema de Theil et al. (Capítulo 8). Mudou-se várias suposições, e Theil não usa controle ótimo para resolver o problema. Porém, as idéias fundamentais do problema permanecem.

Uma empresa fabrica, armazena, e vende um produto. A demanda do produto foi estimado para os próximos dezoito mêses. O produto não é vendido imediatamente, mas fica estocado até que seja comprado. A administração da empresa quer minimizar os custos de fabricação e estocagem do produto.

Tais custos podem ser fixos, ou podem depender da programação da produção. Matéria prima é um custo fixo. Este custo será incorrido independentemente da programação da produção, e não será considerada aqui. Outros custos dependem de variáveis controláveis pela administração. A administração pode admitir e demitir empregados, e pode determinar o nível de produção. A admissão de empregados será representada pela variável h_t. Um valor negativo representa demissão. O nível de produção no período t será representado por p_t.

A admissão/demissão de empregados tem um impacto nas horas extras trabalhadas, ou no tempo ocioso. Uma força de trabalho insuficiente para a produção terá que trabalhar horas extras, que são mais caras do que as horas da jornada normal. No caso oposto, uma força de trabalho grande demais incorrerá tempo ocioso.

Pode-se representar a força de trabalho em período t pela variável q_t. Ainda, pode-se chamar de c_1 a produção média de cada empregado. Então, $c_1 q_t$ representa a produção do período t na jornada de trabalho normal. Porém, duas suposições a mais vão alterar esta equação. Primeiro, supõe-se que o empregado contratado no período t só estará treinado o suficiente para contribuir em período t+1. Além disso, os empregados demitidos têm que ser avisados da sua demissão com um período de antecedência. Então, um empregado cortado em período t-1 trabalhará no período t. Então a produção da jornada normal do período t, pn_t será:

(63) $pn_t = c_1 q_{t-1}$

A folha de pagamento depende do número de empregados e das horas extras que eles trabalham. A folha de pagamento regular é uma função linear do número de empregados. Supondo que c_2 é o custo incorrido para cada empregado trabalhando na jornada normal, a folha de pagamento regular no período t pode ser representada pela Figura 1:

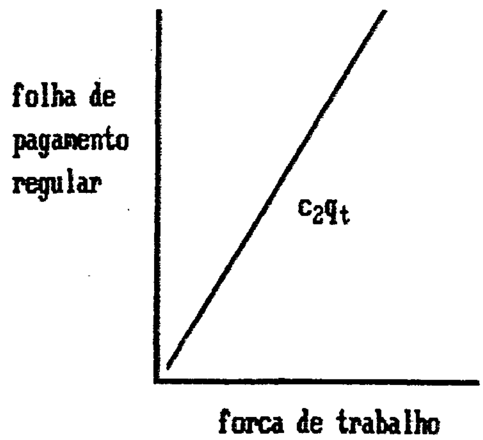

FIGURA 1 - Custos de folha de pagamento regular

O tamanho da força de trabalho é também relevante nos cálculos das horas extras e ociosas. Uma força pequena demais precisará de muitas horas extras para cumprir a produção. Usando a simbologia anteriormente apresentada, horas extras resultam quando $p_t > c_1q_{t-1}$. Tempo ocioso resulta quando a força de trabalho não pode ser utilizado eficientemente, e $p_t < c_1q_{t-1}$. A seguinte expressão expressa horas extras/ociosas em termos de unidades de produto.

$$(64) \quad ot_t = p_t - c_1q_{t-1}$$

Um valor positivo de ot_t implica horas extras. Um valor negativo implica horas ociosas.

Ainda supõe-se que o custo de uma hora extra, c_3, é igual ao custo de uma hora ociosa. Então, o custo de horas extras/ociosas pode ser representado graficamente na forma de um "v", como na Figura 2.

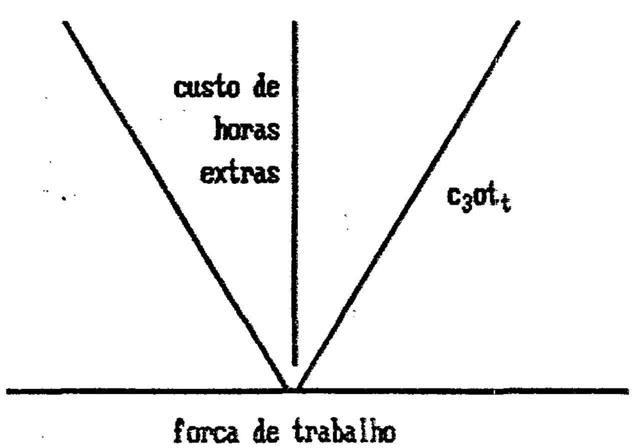

FIGURA 2 - Custo de horas extras/ociosas

Na verdade, o custo de horas extras não é linear. O empregado cansa, e o seu trabalho na última hora extra é menos eficiente do que na primeira hora extra. O custo de cada hora extra incremental é maior do que a anterior. O mesmo acontece com as horas ociosas. Então, uma função quadrática é mais adequada para a representação do custo de horas extras. (Em casos particulares, isto deve ser verificado através de um procedimento de regressão.) A função quadrática é representada na Figura 3.

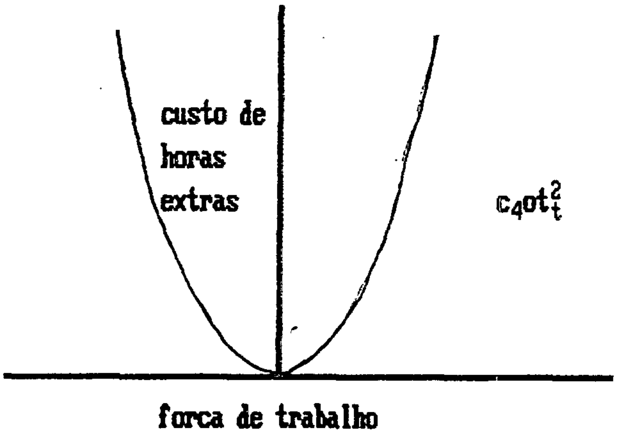

FIGURA 3 - Custo de horas extras/ociosas (funcao quadratica)

Para reduzir os custos fixos e de horas extras da força de trabalho, os empregados podem ser admitidos e demitidos. Esta variável, h_t, liga a força de trabalho em períodos consecutivos.

(65) $q_t = q_{t-1} + h_t$

Em outras palavras, a força de trabalho no final do período t é igual à força de trabalho no final do período t-1 mais (menos) o número de empregados admitidos (demitidos).

A variação na força de trabalho também tem custos. A admissão de um empregado novo resulta em custos preliminares: análise de currículos, entrevistas, testes de aptidão e exames médicos. Além disso, o empregado novo receberá o seu salario integral durante o treinamento, antes que ele tenha começado a contribuir para as rendas da empresa.

A demissão de empregados pode implicar o pagamento de indenização ao empregado demitido, a reorganização da linha de montagem, o sofrimento da reputação da empresa, e mais dificuldade em futuras contratatações de empregados novos. Pode até provocar uma greve.

Tanto no caso de demissão quanto no caso de admissão, os custos incrementais aumentam. Então, semelhante ao caso dos custos de horas extras, os custos de variação na força de trabalho podem ser representados em uma função quadrática:

$$(66) \quad cv_t = c_5 h_t^2$$

Agora considera-se a outra variável diretamente controlável pela administração, a produção. Como se vê na (64), a produção pesa no cálculo nas horas extras, e então indiretamente na admissão/demissão de empregados (66). Neste modelo, também afeta o nível e custos de estoque.

O produto fabricado não sai da fábrica imediatemente, mas fica estocado em um armazém até que seja comprado. O nível de estoque no final de um período t depende do seu nível no final do período anterior t-1, e da produção e das vendas do período t.

$$(67) \quad y_t = y_{t-1} + p_t - s_t$$

Manter o estoque tem custos: custo de manter o armazém, custo de oportunidade do dinheiro investido no estoque, custo de seguros para o estoque, e os custos de seu possível estrago, roubo ou obsolescência. Porém, estoque inadequado tem custos também. Se o cliente cujo pedido não é atendido esperar até que o produto esteja disponível, no mínimo a reputação da empresa vai sofrer. Pior, o cliente pode comprar o produto da concorrência. Se este cliente for permanentemente perdido, a perda pode ser significativa.

Supõe-se que existe um nível ótimo de estoque, que é uma função das vendas antecipadas para os períodos futuros próximos. Para o modelo aqui sendo tratado, considera-se tais níveis como dados, determinados exteriormente. Supõe-se também que y_{t-1} representa o estoque no final do período t-1, y_t o estoque no final do período t, e que y^*_t representa o nível ótimo médio para período t. Então, pode-se estimar o afastamento do estoque do seu nível ótimo durante o período t por:

$$(68) \quad vi_t = \tfrac{1}{2}(y_{t-1} + y_t) - y_t^*$$

$$vi_t = \tfrac{1}{2}(y_{t-1} + y_{t-1} + p_t - s_t) - y_t^*$$

$$vi_t = y_{t-1} + \tfrac{1}{2}p_t - \tfrac{1}{2}s_t - y_t^*$$

Supõe-se também que o afastamento deste nível ótimo tem um custo incrementalmente maior, mas igual para afastamentos positivos ou negativos da mesma distância. Uma estimativa dos custos reais e de oportunidade do afastamento do nivel ótimo de estoque seria:

$$(69) \quad cvi_t = c_6 vi_t^2$$

Agora, pode-se construir um modelo que pode ser resolvido usando controle ótimo. Os custos a serem minimizados são:

folha de pagamento:

normal: $c_2\ q_t$

hora extra: $c_4\ (ot_t)^2$

admissão/demissão: $c_5\ h_t^2$

afastamento do nível desejado de estoque: $c_6\ (vi_t)^2$

que seriam expressos na formulação do problema como:

$$(70) \quad \min \sum_{k=1}^{N} c_4.ot_t^2 + c_6.vi_t^2 + c_5.h_t^2 + c_2.q_t$$

As equações de sistema são:

$$(71)\quad y_t = y_{t-1} + p_t - s_t$$

$$q_t = q_{t-1} + h_t$$

$$ot_t = -c_1 q_{t-1} + p_t$$

$$vi_t = y_{t-1} + \tfrac{1}{2}p_t - \tfrac{1}{2}s_t - y_t^*$$

Onde:

y_t : estoque no final do período t (estado).

p_t : produção durante o período t (controle).

s_t : vendas durante o período t (exógeno).

q_{t-1} : número de empregados produzindo no período t (estado), determinado no período t-1.

h_t : admissão/demissão de empregados durante o período t (controle).

ot_t : estimativa das únidades produzidas usando horas extras, ou não produzidas durante horas ociosas, no período t (estado).

c_1 : produção media de cada empregado em um período (constante).

vi_t : estimativa do afastamento do produto do seu nível ótimo de estoque no período t (estado).

$y*_t$: nível médio de estoque desejado no período t (exógeno).

Agora, expressa-se a função objetivo e as equações de sistema em forma matricial. O vetor de controle tem um impacto imediato. Então, estrutura-se a função objetivo com o vetor de controle embutido no vetor de estado:

(72)

$$\min \ \frac{1}{2}\sum_{t=1}^{N} \begin{bmatrix} y \\ q \\ \text{ot} \\ \text{vi} \\ p \\ h \end{bmatrix}_t \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_5 \end{bmatrix} \begin{bmatrix} y & q & \text{ot} & \text{vi} & p & h \end{bmatrix}_t + \begin{bmatrix} 0 & c_2 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ q \\ \text{ot} \\ \text{vi} \\ p \\ h \end{bmatrix}_t$$

As equações de sistema são:

(73)

$$\begin{bmatrix} y \\ q \\ \text{ot} \\ \text{vi} \\ p \\ h \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -c_1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ q \\ \text{ot} \\ \text{vi} \\ p \\ h \end{bmatrix}_{t-1} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \frac{1}{2} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ h \end{bmatrix}_t + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{2} & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ y^* \end{bmatrix}_t$$

Se fossem exigidos números inteiros para as variáveis empregados e produção, a maneira ideal de resolver este problema seria com programação dinâmica discreta. Desta maneira não apareceria uma solução que mandasse contratar, por exemplo, 2.34 empregados ou produzir 7.34 unidades do produto.

Mesmo se a lógica exigisse números inteiros para a produção e a força de trabalho, os valores da solução poderiam ser arrendondados se fossem grandes. Em uma força de trabalho de 1000 empregados, um mais ou menos não faria grande diferença. Então, 1000.23 pode ser tratado como 1000. Porém, se a solução mandar uma força de trabalho de 2.4 empregados, a utilidade do modelo poderia ser questionada.

Na presente aplicacão. supõe-se que empregados que trabalham tempo parcial podem ser contratados. Então, 2.4 empregados representaria dois que trabalham tempo integral, e um que trabalha só quarenta por cento da jornada de trabalho normal. Além disso, supõe-se que o produto pode ser fabricado em lotes não inteiros.

A solução determinada pelo algoritmo de controle ótimo segue no Quadro 7. A parte algoritmo do programa precisou menos de dez segundos para obter esta solução. A qualidade da solução é boa, se os custos realmente foram lineares e quadráticos, e se resultados não inteiros são desejados ou aceitáveis. Caso estas suposições não forem corretas, então programação dinâmica discreta seria o método mais adequado para o problema. Porém, na maioria dos casos, a obtenção de uma solução usando programação dinâmica levaria bem mais tempo. Programação dinâmica pode exigir muitas tentativas e comparações para determinar a ação ótima em um periodo de tempo.

Os custos das admissões foram \$ 876,555; da folha de pagamento regular, \$ 2,374,719; das unidades produzidas com horas extras, \$ 53,870; e dos níveis não ótimos de estoque, \$ 450,490. A soma destes custos é \$ 3,775,636.

Vetor de Controle

Período	Produção	Admissão
1	2028.244	9.804
2	2203.784	2.487
3	2245.530	4.474
4	2334.983	5.924
5	2454.175	6.239
6	2579.090	6.320
7	2705.385	6.533
8	2835.230	7.880
9	2989.472	13.933
10	3265.693	23.406
11	3751.462	8.166
12	3921.063	-15.853
13	3580.882	-7.499
14	3425.747	8.071
15	3596.190	11.000
16	3819.166	9.759
17	4017.777	3.719
18	4091.065	-0.800

Vetor de Estado

Período	Estoque	Força de Trabalho	Horas Extras	Estoque Extra
0	300.000	100.000	0.000	0.000
1	328.244	109.804	28.244	-99.878
2	432.028	112.291	7.703	-54.864
3	472.559	116.765	-0.292	-4.706
4	492.542	122.689	-0.320	3.550
5	515.717	128.928	0.387	1.130
6	541.807	135.248	0.525	-0.238
7	567.192	141.781	0.422	-0.500
8	588.422	149.661	-0.390	-5.193
9	622.895	163.594	-3.752	-18.342
10	785.588	187.000	-6.195	1.241
11	969.050	195.167	11.457	122.319
12	430.112	179.314	17.728	-78.419
13	752.994	171.815	-5.396	-83.447
14	757.741	179.886	-10.550	47.367
15	761.931	190.885	-1.521	15.836
16	810.096	200.645	1.458	5.014
17	867.874	204.363	4.886	18.985
18	800.939	203.563	0.799	-26.594

QUADRO 7 - Solução do exemplo micro-econômico

4.2.1 Um problema da macro-economia brasileira: formulação:

Na aplicação convencional de controle ótimo, as matrizes de coeficientes são obtidas usando técnicas de regressão. Em Kendrick (1982, p. 26 - 32) apresenta-se um modelo pequeno onde o vetor de estado tem elementos para gastos de consumo pessoal e investimento privado, e o vetor de controle tem um só elemento, o orçamento governamental.

A natureza instável da economia brasileira impossibilitou o desenvolvimento de um modelo desta natureza. Então, tentou-se uma outra abordagem, usando como base uma coleção de equações em Modiano (1988, p. 271). A essência do modelo foi a determinação dos melhores valores de certas variáveis para provocar crescimento na economia brasileira e a redução da dívida externa brasileira. As variáveis a serem determinadas foram os níveis dos preços das importações e exportações brasileiras, e o valor da renda mundial. Eles assumiram o papel das variáveis de controle.

Os resultados deste modelo não foram muito gratificantes. Porém, para ilustrar em mais detalhes como o software é utilizado e como algumas limitacões podem aparecer, uma explicação do modelo e o procedimento usado segue.

O modelo foi construído de seis equações, que se apresentam abaixo. Em cada caso, expressa-se primeiro a equação na sua forma mais "lógica". Daí, usando operações algebráicas transforma-se a equação numa forma mais conveniente para a aplicação de controle ótimo.

A primeira equação expressa a mudança na dívida externa como a diferença das exportacões e importações e os juros incorridos da dívida do período anterior. Neste modelo, exportações, importações, e a dívida externa foram variáveis de estado. A taxa de juros externa foi estimada.

$$(74)\quad X_t - M_t - \bar{r}D_{t-1} = D_{t-1} - D_t$$

$$D_t + X_t - M_t = \bar{r}D_{t-1} + D_{t-1}$$

$$D_t + X_t - M_t = (1+\bar{r})D_{t-1}$$

Onde:

X_t = exportacoes

M_t = importacoes

$\bar{r}$ = taxa de juros externa

D_t = divida externa

A segunda equação expressa as exportações em um período como uma função de três variáveis do período anterior: a renda em outros países, o nível de preço das exportações, e as exportações. Um termo constante (Ax) também aparece na equação. No modelo, renda em outros países e o nível de preços das exportações foram variáveis de controle. Os coeficientes Ax, Bx, Cx e Dx foram estimados.

$$(75)\quad X_t = Ax + BxY^*_{t-1} + CxPx_{t-1} + DxX_{t-1}$$

$$X_t = Ax + DxX_{t-1} + BxY^*_{t-1} + CxPx_{t-1}$$

Onde:

Y^*_t = renda em outros paises

Px_{t-1} = preco exportacoes

A terceira equação tem estrutura semelhante à segunda. Tem-se importações em um período como uma função da renda brasileira, o nível de preço das importações e as importações do período anterior. Um termo constante também aparece na equação. O nível de preços das importações ficou no vetor de controle, e a renda brasileira foi para o vetor de estado. Os coeficientes Am, Bm, Cm e Dm foram estimados.

(78) $I_t = Sp_t + Sg_t + Sx_t - hY_{t-1}$

$Sp_t = Sp\{(1-t)Y_t - \bar{r}D_{t-1}\}$

$Sg_t = tY_t - O_{t-1}$

$Sx_t = M_t - X_t + \bar{r}D_{t-1}$

$I_t = Sp\{(1-t)Y_t - \bar{r}D_{t-1}\} + tY_t - O_{t-1} + M_t - X_t + \bar{r}D_{t-1} - hY_{t-1}$

$X_t - M_t - (Sp(1-t) + t)Y_t + I_t = \bar{r}(1-Sp)D_{t-1} - hY_{t-1} - O_{t-1}$

Onde:

Sp_t = poupanca do setor privado

Sg_t = poupanca do setor publico

Sx_t = poupanca do setor externo

t = aliquota geral de impostos

O_t = orcamento do governo

A sexta equação estima orçamento governamental como 25% da renda no período anterior.

(79) $O_t = .25Y_{t-1}$

Expressando estas equações em forma matricial, tem-se:

(80)
$$\begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -Bm & 0 & 0 \\ 0 & 0 & 0 & 1 & -k & 0 \\ 0 & 1 & -1 & -(Sp(1-t)+t) & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D \\ X \\ M \\ Y \\ I \\ O \end{bmatrix}_t = \begin{bmatrix} 1+\bar{r} & 0 & 0 & 0 & 0 & 0 \\ 0 & Dx & 0 & 0 & 0 & 0 \\ 0 & 0 & Dm & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \bar{r}(1-Sp) & 0 & 0 & -h & 0 & 1 \\ 0 & 0 & 0 & .25 & 0 & 0 \end{bmatrix} \begin{bmatrix} D \\ X \\ M \\ Y \\ I \\ O \end{bmatrix}_{t-1}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ Bx & Cx & 0 \\ 0 & 0 & Cm \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y* \\ Px \\ Pm \end{bmatrix}_{t-1} + \begin{bmatrix} 0 \\ Am \\ Ax \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Que pode ser expressa:

(81) $F^* x_t = A^* x_{t-1} + B^* u_{t-1} + C^*$

Multiplicando a inversa da matriz F* por todos os termos da equação resulta em:

(82) $x_t = (F^*)^{-1} A^* x_{t-1} + (F^*)^{-1} B^* u_{t-1} + (F^*)^{-1} C^*$

Que pode ser expressa na forma padrão:

(83) $x_t = Ax_{t-1} + Bu_{t-1} + C$

Resumindo, o vetor de controle tem os elementos de renda mundial, nível de preços de importações e nível de preços de exportações. O vetor de estado inclui dívida externa, exportações, importações, renda brasileira, investimento, e orçamento governamental.

Então, tentou-se determinar valores dos elementos do vetor de controle que favoreceriam a redução da divida externa e o crescimento da economia. Contudo, a solução obtida não foi interessante, e sua implantação seria impraticável, dado que as variáveis, definidas como elementos de controle dentro do modelo, efetivamente não podem ser manipulados pelo decisor interessado na solução do problema, isto é, o governo brasileiro.

Nesta primeira tentativa de modelagem, aplicou-se uma penalidade positiva à dívida externa e uma penalidade negativa à renda nacional. Deixou-se os outros elementos sem penalidade.

4.2.2 Dados, Modificações e Resultados:

Na procura de resultados úteis, foram feitas várias modificações na formulação descrita na seção anterior. Aqui descreve-se as barreiras que encontrou-se nesta procura.

Na primeira tentativa, associou-se os valores inicias do Quadro 8 aos elementos do vetor de estado :

$$D_0 = \$\ 120,000,000,000$$
$$X_0 = 30,000,000,000$$
$$M_0 = 15,000,000,000$$
$$Y_0 = 360,000,000,000$$
$$I_0 = 90,000,000,000$$
$$O_0 = 90,000,000,000$$

QUADRO 8 - Condicoes inicias do vetor de estado

Aplicou-se os valores das constantes no Quadro 9:

$\bar{r} = 0.05$

$Bm = 0.05$

$k = 1$

$h = 0.05$

$Sp = 0.05$

$t = 0.15$

$Dx = 0.1$

$Dm = 0.1$

$Bx = 0.003$

$Cx = -24{,}000{,}000{,}000$

$Cm = -\ 4{,}500{,}000{,}000$

$Am = 0$

$Ax = 0$

QUADRO 9 - Valores das constantes do modelo macro-econômico

Que resultou nas seguintes matrizes de F*, A*, e B*:

(84)

$$F^* = \left|\begin{matrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.05 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & -0.1925 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}\right|$$

(85)

$$A^* = \left|\begin{matrix} 1.05 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.0475 & 0 & 0 & -0.05 & 0 & 1 \\ 0 & 0 & 0 & 0.25 & 0 & 0 \end{matrix}\right|$$

(86)

$$B^* = \left|\begin{matrix} 0 & 0 & 0 \\ 0.003 & -2.40E+10 & 0 \\ 0 & 0 & -4.50E+09 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}\right|$$

Calculou-se a inversa da matriz F* como:

(87)

$$F^{*-1} = \begin{vmatrix} 1.000 & -1.066 & 1.066 & 0.066 & 0.066 & 0.000 \\ 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.066 & 1.066 & 0.066 & 0.066 & 0.000 \\ 0.000 & -1.320 & 1.320 & 1.320 & 1.320 & 0.000 \\ 0.000 & -1.320 & 1.320 & 0.320 & 1.320 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 \end{vmatrix}$$

Quando pre-multiplica-se A* e B* por F^{*-1}, obtem-se A e B:

(88)

$$A = \begin{vmatrix} 1.053 & -0.107 & 0.107 & 0.063 & 0.000 & 0.066 \\ 0.000 & 0.100 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.003 & -0.007 & 0.107 & 0.063 & 0.000 & 0.066 \\ 0.063 & -0.132 & 0.132 & 1.254 & 0.000 & 1.320 \\ 0.063 & -0.132 & 0.132 & 0.254 & 0.000 & 1.320 \\ 0.000 & 0.000 & 0.000 & 0.250 & 0.000 & 0.000 \end{vmatrix}$$

(89)

$$B = \begin{vmatrix} -0.003 & 2.56E+10 & -4.80E+09 \\ 0.003 & -2.40E+10 & 0.00E+00 \\ 0.000 & 1.58E+09 & -4.80E+09 \\ -0.004 & 3.17E+10 & -5.94E+09 \\ -0.004 & 3.17E+10 & -5.94E+09 \\ 0.000 & 0.00E+00 & 0.00E+00 \end{vmatrix}$$

Na seleção de matrizes de penalidade, usou-se apenas penalidades lineares. Penalidades quadráticas penalizariam igualmente valores positivos e negativos, não percebendo que uma "renda" de \$10 é melhor do que de -\$10, ou que uma "dívida" de \$10 é pior do que -\$10.

Penalidades quadráticas do vetor de estado:

(90)

$$W = \begin{vmatrix} 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{vmatrix}$$

Penalidades lineares do vetor de estado:

(91)

$$
w = \begin{vmatrix} 10.000 \\ 0.000 \\ 0.000 \\ -10.000 \\ 0.000 \\ 0.000 \end{vmatrix}
$$

Penalidades quadráticas do vetor de controle:

(92)

$$
L = \begin{vmatrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{vmatrix}
$$

Penalidades lineares do vetor de controle:

(93)

$$
l = \begin{vmatrix} 0.000 \\ 0.000 \\ 0.000 \end{vmatrix}
$$

Com a obtenção destes dados, o problema pode ser, então, resolvido. O QLP foi executado, e identificou o problema como sendo impossível de resolver. A matrix THETA calculada, usando os dados, não foi inversível. Então, mudou-se a matriz de penalidade W para permitir a inversão de THETA. Para enfatizar crescimento e a redução da dívida, as suas penalidades lineares foram aumentadas.

(94)

$$
W = \begin{vmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 \end{vmatrix}
$$

Penalidades lineares novas do vetor de estado:

(95)

$$w = \begin{vmatrix} 1000.000 \\ 0.000 \\ 0.000 \\ -1000.000 \\ 0.000 \\ 0.000 \end{vmatrix}$$

O problema com as novas penalidades foi executável. Porém, os resultados demonstraram que a formulação do modelo não foi válida. Com estas penalidades, as variáveis não ficaram presas em valores consistentes com a realidade. Por exemplo, o Brasil em um período foi de país devedor de cento vinte bilhões de dólares para um país credor. A renda mundial ficou quase 0 durante 20 períodos, e apareceram orçamentos, e níveis de investimento e preços negativos. Nos Quadros 10 e 11 mostram-se os resultados:

Período	Dívida Externa	Exportações	Importações
0	120000000000	30000000000	15000000000
1	-389527766754	257763883377	-257763883377
2	-522841642340	56918743624	-56918743624
3	-471735862334	-38623931061	38623931061
4	-387653545435	-53834555007	53834555007
5	-307770630393	-49632796157	49632796157
6	-241118651402	-41020255255	41020255255
7	-187907845526	-32633369223	32633369223
8	-146152455069	-25575391366	25575391366
9	-113620406095	-19919835863	19919835863
10	-88359209726	-15471108336	15471108336
11	-68784638045	-11996266083	11996266083
12	-53647347282	-9288261332	9288261332
13	-41974360353	-7177677146	7177677146
14	-33013541671	-5529768350	5529768350
15	-26188026602	-4238096076	4238096076
16	-21062021551	-3217703190	3217703190
17	-17320055233	-2397533697	2397533697
18	-14770705600	-1707676197	1707676197
19	-13386312539	-1061464170	1061464170
20	-13774045817	-140791174	140791174

Período	Renda Brasileira	Investimento	Orçamento Governamental
0	360000000000	90000000000	90000000000
1	-96381135299	-456381135299	90000000000
2	-165823077032	-69441941733	-24095283824
3	-160017737329	5805339702	-41455769258
4	-134007384726	26010352602	-40004434332
5	-107070465531	26936919194	-33501846181
6	-83959607269	23110858262	-26767616382
7	-65282527286	18677079983	-20989901817
8	-50505439837	14777087448	-16320631821
9	-38889621740	11615818096	-12626359959
10	-29753873967	9135747773	-9722405435
11	-22530174611	7223699356	-7438468491
12	-16758987016	5771187595	-5632543652
13	-12069939345	4689047671	-4189746754
14	-8161528852	3908410492	-3017484836
15	-4783762628	3377766223	-2040382213
16	-1725673799	3058088828	-1195940657
17	1187975981	2913649781	-431418449
18	4074066257	2886090276	296993995
19	6920961939	2846895681	1018516564
20	8964907628	2043945688	1730240484

QUADRO 10 - Vetores de estado - Modelo Macro-econômico 1

Período	Renda mundial	Nível de preços de exportações	Nível de preços de importações
0	-0.000	-10.61516	56.54329
1	-0.000	-1.29760	5.07804
2	0.000	1.84649	-11.62593
3	0.000	2.08217	-12.59390
4	0.000	1.84372	-11.02286
5	0.000	1.50237	-8.94555
6	-0.000	1.18881	-7.06566
7	0.000	0.92967	-5.51941
8	0.000	0.72343	-4.29040
9	-0.000	0.56163	-3.32596
10	-0.000	0.43538	-2.57237
11	0.000	0.33703	-1.98369
12	0.000	0.26037	-1.52274
13	0.000	0.20050	-1.16002
14	0.000	0.15355	-0.87207
15	0.000	0.11641	-0.64004
16	0.000	0.08649	-0.44808
17	0.000	0.06116	-0.28094
18	0.000	0.03711	-0.12103
19	0.000	0.00144	0.09191

QUADRO 11 - Vetores de controle - Modelo Macro-econômico 1

Então, tentou-se re-formular o problema como um do tipo "tracking", com valores desejados especificados para todos os vetores. Para determinar os valores desejados dos elementos de estado no período 1, os valores iniciais foram multiplicados por fatores de crescimento. Usou-se tais fatores multiplicados pelos valores do período 1 para calcular os valores do período 2, e assim por diante. Os valores escolhidos pelos fatores dependeram da meta com respeito ao elemento correspondente. Em geral, desejou-se uma gradual diminuição na dívida externa, um gradual crescimento na renda nacional, e estabilidade nos outros vetores. Então, escolheu-se os fatores listados no Quadro 12.

dívida externa	0.95
exportações	1.03
importações	1.03
renda nacional	1.06
investimento	1.03
orçamento governamental	1.03

QUADRO 12 - Fatores de crescimento dos vetores de estado

No caso dos elementos de controle, tentou-se aplicar o mesmo raciocínio. Porém, primeiro precisou-se de valores inicias. Estimou-se o valor inicial do renda mundial em dez trilhões de dólares, ou seja $10,000,000,000,000. Não foram disponíveis valores inicias para os níveis de preços das importações e exportações. Então, tentou-se calculá-los a partir das equações de sistema. Para o problema atual, tem-se:

(96)

$$
\left|\begin{matrix} D \\ X \\ M \\ Y \\ I \\ O \end{matrix}\right|_{t+1} = \left|\begin{matrix} 1.053135 & -0.10660 & 0.106600 & 0.062706 & 0 & 0.066006 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0.003135 & -0.00660 & 0.106600 & 0.062706 & 0 & 0.066006 \\ 0.062706 & -0.13201 & 0.132013 & 1.254125 & 0 & 1.320132 \\ 0.062706 & -0.13201 & 0.132013 & 0.254125 & 0 & 1.320132 \\ 0 & 0 & 0 & 0.25 & 0 & 0 \end{matrix}\right| \left|\begin{matrix} D \\ X \\ M \\ Y \\ I \\ O \end{matrix}\right|_{t}
$$

$$
+ \left|\begin{matrix} -0.00319 & 2.6E+10 & -4.8E+09 \\ 0.003 & -2.4E+10 & 0 \\ -0.00019 & 1.6E+09 & -4.8E+09 \\ -0.00396 & 3.2E+10 & -5.9E+09 \\ -0.00396 & 3.2E+10 & -5.9E+09 \\ 0 & 0 & 0 \end{matrix}\right| \left|\begin{matrix} Y^* \\ Px \\ Pm \end{matrix}\right|_{t}
$$

De um período para o proximo, supõe-se mudanças graduais. Então, pensou-se que seria possível de estimar Px e Pm através da substituição dos valores iniciais do vetor de estado e a renda mundial. Então, valores de Px e Pm que provacassem uma leve variação no vetor de estado no período seguinte seriam escolhidos. Substituindo:

(97)

$$\begin{vmatrix} D\\ X\\ M\\ Y\\ I\\ O \end{vmatrix}_{t+1} = \begin{vmatrix} 1.053135 & -0.10660 & 0.106600 & 0.062706 & 0 & 0.066006\\ 0 & 0.1 & 0 & 0 & 0 & 0\\ 0.003135 & -0.00660 & 0.106600 & 0.062706 & 0 & 0.066006\\ 0.062706 & -0.13201 & 0.132013 & 1.254125 & 0 & 1.320132\\ 0.062706 & -0.13201 & 0.132013 & 0.254125 & 0 & 1.320132\\ 0 & 0 & 0 & 0.25 & 0 & 0 \end{vmatrix} \begin{vmatrix} 1.2E+11\\ 3.0E+10\\ 1.5E+10\\ 3.6E+11\\ 9.0E+10\\ 9.0E+10 \end{vmatrix}_{t}$$

$$+ \begin{vmatrix} -0.00319 & 2.6E+10 & -4.8E+09\\ 0.003 & -2.4E+10 & 0\\ -0.00019 & 1.6E+09 & -4.8E+09\\ -0.00396 & 3.2E+10 & -5.9E+09\\ -0.00396 & 3.2E+10 & -5.9E+09\\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 1.0E+13\\ Px\\ Pm \end{vmatrix}_{t}$$

Porém, achou-se que não existem valores de Px e Pm que causam tal variação paulatina no vetor de estado. O valores de Px e Pm que resultaram em mais estabilidade no vetor de estado de um período para o próximo provocaram uma mudanca média de 22.56%, como mostra-se no Quadro 13.

	CONTROLE PER 0		ESTADO PER 0	ESTADO PER 1	DELTA	|DELTA|
Y*	1.0E+13	D	1.200E+11	1.110E+11	-7.54%	7.54%
Px	0.12	X	3.000E+10	3.012E+10	0.40%	0.40%
Pm	2.8	M	1.500E+10	1.507E+10	0.47%	0.47%
		Y	3.600E+11	5.234E+11	45.39%	45.39%
		I	9.000E+10	1.634E+11	81.56%	81.56%
		O	9.000E+10	9.000E+10	0.00%	0.00%
					|Delta|/6 =	22.56%

QUADRO 13 - Efeito do vetor de controle no vetor de estado (1)

Outras tentativas de achar estabilidade aparecem nos Quadros 14 e 15. No Quadro 14, tentou-se reduzir a mudança no elemento de investimento. Achou-se que ao reduzir a mudança porcentual de investimento, os outros elementos fugiram de controle.

CONTROLE PER 0			ESTADO PER 0	ESTADO PER 1	DELTA	¦ DELTA ¦
Y*	1.0E+13	D	1.200E+11	5.191E+10	-56.74%	56.74%
Px	0.85	X	3.000E+10	1.260E+10	-58.00%	58.00%
Pm	19	M	1.500E+10	-6.149E+10	-509.90%	509.90%
		Y	3.600E+11	4.503E+11	25.08%	25.08%
		I	9.000E+10	9.030E+10	0.33%	0.33%
		O	9.000E+10	9.000E+10	0.00%	0.00%
					¦Delta¦/6 =	108.34%

QUADRO 14 - Efeito do vetor de controle no vetor de estado (2)

CONTROLE PER 0			ESTADO PER 0	ESTADO PER 1	DELTA	¦ DELTA ¦
Y*	1.0E+13	D	1.200E+11	1.421E+11	18.42%	18.42%
Px	1	X	3.000E+10	9000000000	-70.00%	70.00%
Pm	1	M	1.500E+10	2.510E+10	67.33%	67.33%
		Y	3.600E+11	5.620E+11	56.11%	56.11%
		I	9.000E+10	2.020E+11	124.42%	124.42%
		O	9.000E+10	9.000E+10	0.00%	0.00%
					¦Delta¦/6 =	56.05%

QUADRO 15 - Efeito do vetor de controle no vetor de estado (3)

A não existência de valores de Px e Pm que resultam em valores consecutivos de vetores de estado estáveis, pôs em dúvida a utilidade do modelo.

Também deve-se mencionar outro defeito da abordagem usada. Na abordagem convencional, são atribuídas penalidades aos elementos de controle consistentes com os seus custos reais. Convencionalmente, o orçamento governamental é um elemento de controle, e a penalidade associada está fundamentada na realidade. E lógico associar uma penalidade com este elemento; além disso, tal penalidade é ligada à realidade. Porém, não é assim nos elementos de controle no modelo macro-econômico aqui desenvolvido. Não pode-se "penalizar" a renda mundial ou os níveis de preços de importações e exportações na mesma maneira.

Então, além de limitações impostas pelas estimativas e simplificações inadequadas, a própria idéia de aplicar controle ótimo nas condições descritas não era legítima.

Mesmo assim, muitas tentativas de achar valores consistentes com a realidade foram tentadas. Afinal, o modelo não deu certo.

CAPÍTULO 5: CONCLUSÕES

Neste trabalho foi apresentada uma contribuição ao problema QLDD, para qual é desenvolvido um software. A sua documentação e os comentários dentro do programa são escritos em Português. O software resolve tanto o problema geral quanto o problema de "tracking", e pode trabalhar com vetores exógenos. Outras habilidades incluem a identificação de problemas não resolviveis, a seleção de saída, o cálculo de multiplicadores, opções de digitar, confirmar e mudar dados, e a apresentação gráfica dos resultados na tela.

A aplicação do software a um problema de controle de produção e admissão de empregados forneceu resultados satisfatórios. A sua aplicação à economia brasileira foi decepcionante. A natureza instável da economia nacional, com o seu forte componente externo, impossibilitou o uso convencional de controle ótimo em um ambiente macro-econômico. Tentou-se uma abordagem alternativa, mas não resultou em resultados úteis.

O fato do software estar disponível em código fonte permite a sua fácil adaptação para problemas especiais. Além disso, o software pode servir como uma base de um programa que resolve problemas com equações não lineares ou problemas estocásticos. Talvez com tais habilidades a economia brasileira poderia ser modelada com mais successo.

BIBLIOGRAFIA

Chow, Gregory, Analysis and Control of Dynamic Economic Systems, 1975, John Wiley & Sons, Inc., New York.

Fritsch, Winston e Modiano, Eduardo Marco, "A restrição externa ao crescimento econômico brasileiro: uma perspectiva de longo prazo", Pesquisa e Planejamento Econômico, agosto 1988, Rio de Janeiro.

Handa, V.K. e Bárcia, R.M., "Construction Production Planning", Journal of Construction Engineering and Management, junho 1986.

Hastings, N. A. J., Dynamic Programming With Management Applications, 1973, The Butterworth Group, London.

Kendrick, David A., Stochastic Control for Economic Models, 1982, McGraw Hill, New York.

Kendrick, David A. e Vila, F., "QLP: a Program for Deterministic Quadratic Linear Control Problems" (working papers), 1988, University of Texas Department of Economics.

Samohyl, Robert W., Anotações de aula - Macro-economia, Florianópolis 1988.

Swan, Tom, Mastering Turbo Pascal 5.5, 1989, Hayden Books, Indianapolis.

Theil, et. al., Operations Research and Quantitative Economics, 1965, McGraw Hill, New York.

T. McCutchen
PQL
Dissertação de Mestrado
UFSC/EPS
1990

APENDICE I: MANUAL DO PROGRAMA PQL

1.0 Introdução:

O PQL é um programa de Turbo Pascal 5.0 para resolver problemas quadráticos lineares determísticos discretos (QLDD) de controle ótimo. Resolve o problema geral, e também o caso especial de "tracking". Este apêndice, o manual do programa, foi feito com o intuito de ser o mais auto-suficiente possível. Não é necessario consultar as outras partes da dissertação se não houver interesse nas demonstrações matemáticas, as aplicações e outras informações detalhadas.

A primeira parte do manual, Seção 1.1, descreve matemáticamente o tipo de problema para qual desenvolveu-se o PQL. Na Seção 1.2, tem-se uma breve explicação de como executar o programa. Na Seção 1.3 descreve-se as estruturas do arquivo de dados.

Na Seção 1.4 trata-se das unidades (partes) do programa com uma descrição breve de cada unidade, e nos casos onde a programação é mais complicada oferece-se explicações mais rigorosas. Esta seção é de mais interesse para quem quer entender como o programa funciona, ou até modificá-lo.

A última parte, Seção 1.5, descreve o uso e a programação do pacote gráfico, PLOTTER, que pode ser chamado depóis que PQL for executado.

1.1 Definição do problema:

O PQL resolve problemas quadráticos lineares determísticos discretos de controle ótimo da forma:

Achar u_k para $k = 0, 1, \dots, N-1$

(1) $\text{Min } J = \frac{1}{2}x_N' W_N x_N + w\__N' x_N + \sum_{k=0}^{N-1} (\frac{1}{2}x_k' W_k x_k + w\__k' x_k + \frac{1}{2}u_k' L_k u_k + l\__k' u_k)$

Sujeito a:

(2) $x_{k+1} = Ax_k + Bu_k + C + Dz_k$

para $k = 0, 1, \dots, N-1$

com x_0 dado

Onde:

x_k é um vetor de estado de ST elementos.

u_k é um vetor de controle de CON elementos.

z_k é um vetor exógeno de EX elementos.

W_k é uma matriz de penalidades (ST x ST).

$w\__k$ é um vetor de penalidade de ST elementos.

L_k é uma matriz de penalidades (CON x CON).

$l\__k$ é um vetor de penalidade de CON elementos.

A_k é uma matriz (ST x ST).

B_k é uma matriz (ST x CON).

C_k é um vetor (ST x 1).

D_k é uma matriz (ST x EX).

N é o número de períodos no problema.

Esta forma é a forma geral. Também, o PQL resolve um caso particular importante, o problema de "tracking". Neste caso a forma é:

Achar u_k $k = 0, 1, \ldots, N-1$ para

(3) $$\text{Min } J = \tfrac{1}{2}[x_N - \bar{x}_N]'W_N[x_N - \bar{x}_N] + \tfrac{1}{2}\sum_{K=0}^{N-1}([x_N - \bar{x}_N]'W_N[x_N - \bar{x}_N] + [u_N - \bar{u}_N]'L_N[u_N - \bar{u}_N])$$

sujeito a:

(4) $$x_{K+1} = Ax_K + Bu_K + C + Dz_K$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

$\bar{x}_k$ dado para $k = 0, 1, \ldots, N$

$\bar{u}_k$ dado para $k = 0, 1, \ldots, N-1$

Onde:

$\bar{x}_k$ é um vetor de ST elementos dos valores desejados para o vetor de estado.

$\bar{u}_k$ é um vetor de CON elementos dos valores desejados para o vetor de controle.

As demais matrizes são como definidas para o caso geral.

O algoritmo que o PQL usa para solucionar o problema é derivado no Capítulo 3.

1.2 Como executar o PQL:

Para executar o programa o usuario deve primeiro carregar a linguagem Turbo Pascal 5.0. A seguir, a opção "File,Load" deve ser selecionada do menu do Turbo Pascal, e o nome do arquivo "PQL" digitado. Para começar a execução, a opção "Run" deve ser escolhida. Então, aparece a mensagem:

Como deseja fornecer os dados de entrada?
1. Arquivo de Dados
2. Teclado

Caso a primeira opção for escolida, tem-se que fornecer o nome do arquivo de dados, que é o assunto da Seção 1.3. Para a segunda opção, aparecem instruções detalhadas para digitar os dados.

Depóis que os dados de entrada foram recebidos pelo programa, o PQL apresenta a mensagem:

Ver / Mudar / Continuar / Terminar
1. Ver Dados
2. Mudar Dados
3. Continuar Execucao
4. Terminar Execucao

A primeira opção pode ser chamada para confirmar que o programa recebeu os dados corretamente. Se não foi o caso, é os dados foram lidos de um arquivo de dados, as correções podem ser feitas tanto no arquivo de dados quanto através da segunda opção deste menu, "Mudar Dados".

Dados incorretos muitas vezes resultam de um formato errado no arquivo de dados. Se muitos elementos das matrizes estiveram errados, então deve-se consultar este manual novamente para ver se o arquivo de dados obedeceu as regras de formato. Porém, se existirem alguns erros isolados de datilografia, os dados vindos de arquivo de dados podem ser corregidos usando opção dois. Se esta opção for escolhida, o PQL oferece a opção de gravar os novos dados no arquivo de dados, apagando os valores velhos. Depóis de mandar o program continuar, apareceria na tela:

Deseja gravar as mudanças no arquivo original de dados (S/N) ?

No caso de dados digitados diretamente na execução do programa, a correção deve ser feita com a opção "Mudar Dados". Depóis de escolher a opção "Continuar Execução", o programa grava os dados em um arquivo de dados automaticamente.

Na execução do program, aparecem mensagens na tela avisando o usuario do progresso na resolução do problema. Quando termina, os resultados podem ser vistos no arquivo de saída especificado.

1.3 Dados de entrada em PQL:

1.3.0 Introdução:

Existem duas maneiras de fornecer dados para PQL: ou digitando os dados durante a execução do programa, ou colocando eles de ante-mão em um arquivo de dados. A não ser que o problema seja muito pequeno, é aconselhável o uso de um arquivo de dados para fornecer dados ao PQL.

O conteúdo do arquivo de dados depende do tipo de problema sendo resolvido. Os dados dos 21 ítens iniciais são iguais, não importando se o problema for o problema geral ou o problema de "tracking". Depóis, nos ítens 22 até 24 o problema geral precisa dos valores dos vetores de penalidades l_ e w_ (que são calculados pelo próprio programa no problema de "tracking"). Para os ítens 22 até 24 no problema de "tracking" os dados são os valores desejados dos vetores de estado e controle. Finalmente, itens 25 (Z) e 26 (x_0, condições iniciais) são iguais para ambos os casos.

Em qualquer caso, a colocação dos dados deve seguir rigorosamente as regras explicitadas. Um exemplo de um arquivo de dados para um problema de "tracking" é apresentado em Figura 1. O texto itálico não faz parte do arquivo, mas sim apenas serve para identificar a estrutura do arquivo.

O problema de "tracking" em Figura 1 tem dois elementos no vetor de estado (Consumo e Investimento), um elemento no vetor de controle (Orçamento Governamental), e não tem elementos exógenos. O horizonte é de sete períodos. Pelos valores dos "switches" de impressão, o arquivo de saída detalhada (KENDRICK.DET) receberá os dados do problema e as matrizes Ricatti. Além disso, o arquivo KENDRICK.MUL receberá todos os multiplicadores.

```
--------------------------------------------------------------------
EXEMPLO MACRO-ECONOMICO DO KENDRICK  nome do problema
1 0 0 0 0 0 0 0 0 1 1 1 1 1 1        "switches" de impressão
11 3                                 formato da saída
1                                    "switch" para nomes de variáveis
KENDRICK.DET                         arquivo de saída detalhada
KENDRICK.SUM                         arquivo de saída resumida
KENDRICK.GRA                         arquivo de saída para gráficos
KENDRICK.MUL                         arquivo de saída para multiplicadores
1                                    "switch" do tipo de problema
2                                    número de elementos no vetor de estado
1                                    número de elementos no vetor de controle
0                                    número de elementos no vetor exógeno
7                                    número de períodos
Consumo                              {
Investimento                         { Nomes de elementos de estado
Orcamento Governamental              Nome de elemento de controle
1.014  0.002                         {
0.093  0.753                         { matriz A
-0.004                               {
-0.100                               { vetor B
-1.312                               {
0.448                                { vetor C
1.0 0.0                              {
0.0 1.0                              { W para período 0, 1,..., N-1
100.0   0.0                          {
  0.0 100.0                          { W para período N
  1.0                                { L para período 0, 1, ..., N-1
460.1                                                  {
113.1                                                  { XOBAR
463.551 467.027 470.530 474.059 477.615 481.197 484.806    {
113.948 114.803 115.664 116.531 117.405 118.286 119.173    { XBAR
153.644 154.796 155.957 157.127 158.305 159.492 160.689    { UBAR
460.1                                                  {
113.1                                                  { X0
--------------------------------------------------------
```

FIGURA 1 - EXEMPLO DE ARQUIVO DE DADOS
("Tracking", sem variáveis exógenas)

Um resumo dos elementos de entrada segue:

1.3.1: Nome do problema. Até 50 caráteres incluindo números, letras, símbolos e espaços em branco.

1.3.2: "Switches" de controle de impressão. Nesta linha indica-se a impressão optativa dos dados de entrada. Além disso, existem várias matrizes intermediárias que o programa calcula na determinação das matrizes Ricatti, que opcionalmente pode-se dirigir ao arquivo de saída detalhada. Veja Capítulo 3 para informações detalhadas sobre as matrizes de Ricatti. Finalmente, pode-se solicitar o cálculo e impressão de vários multiplicadores, que explica-se na Seção 2.6.

Todos os quinze "switches" são necesários e terão que ficar na segunda linha.

"Switch"	Controle
1	dados de entrada
2	PHI
3	THETA
4	THETA_INV
5	PSI
6	phi_
7	theta_
8	G
9	g
10	K
11	p
12	multiplicador de impacto
13	multiplicador defasado
14	multiplicador de prazo intermediário
15	multiplicador de muito longo prazo

1.3.3: Na linha que vem imediatemente depóis dos "switches" de impressão, deve-se colocar dois números inteiros, separados por pelo menos um espaço. O primeiro indica quantos espaços os números ocupam na saída (Places). O segundo corresponde ao número de algarismos depóis do ponto decimal (Dec). A seguir apresentam-se alguns exemplos:

O número	em formato Places	Dec	sai
150.4538	10	5	150.45340
150.4538	10	3	150.454
150.4538	10	2	150.45

Controla-se o número de colunas da saída através da constante OutWidth na unidade GLOBAL (ver Seção 1.4.6). O valor default é cinco. Dado que a largura normal de uma folha padrão é de 65 espaços, isso implica que um valor de treze para a variável Places enche a folha (13 x 5 = 65).

1.3.4: "Switch" dos nomes dos elementos dos vetores. Se apenas os números das variáveis forem suficientes para identificá-los, deve-se colocar o valor 0. Para ativar a opção de dar nomes aos elementos, deve-se colocar o valor 1. Neste caso, também deve-se colocar os nomes dos elementos no lugar especificado no ítem 1.3.14.

1.3.5: Nome do arquivo de saída detalhada. Determina-se o conteúdo deste arquivo pelos "switches" da Seção 1.3.2. Deve-se notar que se muitas opções de impressão forem escolhidas, o arquivo detalhado pode assumir um tamanho realmente enorme. Neste caso, para ser lido no computador, recomenda-se o uso de um processador de texto com habilidade de trabalhar com arquivos grandes, como WordStar 4.0. Turbo Pascal 5 trunca os arquivos depóis de um determinado tamanho, só permitindo a examinação da primeira parte da saída.

Como é também o caso nos nomes dos outros arquivos a seguir (itens 1.3.6 até 1.3.8), o nome do arquivo deve obedecer as regras do DOS. Além disso, deve-se ter nomes diferentes para todos estes quatro arquivos. Não é possível, por exemplo, colocar as matrizes dos multiplicadores no arquivo de saída detalhada pelo uso do mesmo arquivo para os dois. Se for desejado que um relatorio contenha informações dos dois arquivos, pode-se juntar eles posteriormente usando um processador de texto.

Recomenda-se que os nomes dos arquivos tenham nomes significativos e semelhantes. Para um problema identificado como "PIN," pode-se ter "DADOS.PIN" como o arquivo fonte de dados, e "SUM.PIN", "DET.PIN", "GRAPH.PIN" e "MULT.PIN" como os arquivos de saída.

1.3.6: Nome do arquivo de saída resumida. Este arquivo conterá os valores de X e U para todos os períodos no problema, e também o valor da função objetivo. No caso de "tracking", também apresentará os valores de XBAR e UBAR.

1.3.7: Nome do arquivo de dados do programa gráfico. Quando executa-se o programa gráfico separado, o nome deste arquivo é solicitado.

1.3.8: Nome do arquivo de saída para os multiplicadores. Mesmo que os multiplicadores não forem pedidos no item 1.3.2, um nome deve ser colocado.

1.3.9: O "switch" do tipo do problema. Colocar 1 para o problema de "tracking" e 0 para o problema geral.

1.3.10: O número de elementos do vetor de estado (ST) do problema.

1.3.11: O número de elementos do vetor de controle (CON) do problema.

1.3.12: O número de elementos no vetor exógeno (EX) do problema.

1.3.13: O número de períodos (N) no problema.

1.3.14: Os nomes dos elementos dos vetores, caso ítem 1.3.4 (o "switch" dos nomes dos elementos) for 1, um nome por linha. Primeiro, coloca-se os ST nomes do vetor de estado. Segue-se com os CON nomes dos elementos de controle. Finalmente coloca-se os nomes dos elementos exógenos, caso eles existirem. Se o ítem 1.3.4 for 0, ítem 1.3.15 deve imediatamente seguir ítem 1.3.13.

1.3.15: A matriz A das equações de sistema, que tem dimensão ST por ST. Deve-se separar os elementos de cada linha da matriz por pelo menos um espaço. Se a matriz A tiver mais que oito colunas, então deve-se partir a matriz e colocar as partições uma em cima da outra. Se a matriz A for 21 por 21, então deve-se colocar ela no arquivo de dados na seguinte maneira:

A -- partição --> [A1 ¦ A2 ¦ A3]

Colocação no arquivo de dados:

A1
A2
A3

Onde

A1 é uma matriz 25 x 8 e representa colunas 1 - 8 de A.

A2 é uma matriz 25 x 8 e representa colunas 9 - 16 de A.

A3 é uma matriz 25 x 5 e representa colunas 17 - 21 de A.

Note que evita-se linhas em branco entre as partições da matriz. Também, não deve-se deixar linhas em branco entre as matrizes.

Aplicam-se as mesmas regras nos casos das matrizes B, D, W, Wn, L, XBAR, UBAR, e Z.

1.3.16: A matriz B das equações de sistema, de dimemsão ST por CON. Valem as regras da Seção 1.3.15.

1.3.17: O vetor C das equações de sistema, de dimensão ST por 1. Colocar como uma coluna de números.

1.3.18: A matriz D das equações de sistema, de dimensão ST por EX. Se o problema não incluir um vetor exógeno, a matriz D não existe e a matriz W deve seguir diretamente o vetor C. Valem as regras da Seção 1.3.15.

1.3.19: A matriz de penalidade W_k da função objetivo, de dimensão ST por ST.

As penalidades desta matriz aplicam aos períodos 0 até N-1. Valem as regras da Seção 1.3.15.

1.3.20: A matriz W_N de penalidade da função objetivo, de dimensão ST por ST. As penalidades de W_N aplicam apenas ao período final. Valem as regras da Seção 1.3.15.

1.3.21: A matriz de penalidade L da função objetivo, de dimensão CON por CON. Valem as regras da Seção 1.3.15.

1.3.22: Este ítem depende do tipo de problema.

Caso geral: O vetor w_k de penalidades da função objetivo, dos vetores de estado nos períodos k = 0, 1, ..., N-1. O w_k tem ST elementos.

Caso de "tracking": O vetor XBAR[I,0] (I = 1, ... , ST) que contem os ST valores desejados do vetor de estado no período zero. Geralmente o XBAR0 será igual ás condições iniciais do vetor de estado (Seção 1.3.26). XBAR[I,0] tem dimensão ST por 1 e deve ser colocado em uma coluna de números.

1.3.23.: Este ítem depende do tipo do problema.

Caso geral: O vetor w_N de penalidades da função objetivo do vetor de estado no último período N. O w_N tem ST elementos.

Caso de "tracking": Uma matriz XBAR, de dimensão ST por N. Cada coluna tem os valores desejados do vetor de estado em um período. As colunas são para os períodos 1, 2, ..., N. Valem as regras da Seção 1.3.15.

1.3.24: Este ítem depende do tipo de problema.

Caso geral: O vector 1_k de penalidades da função objetivo dos vetores de controle nos períodos 0, 1, ... N-1. O u_k tem CON elementos.

Caso de "tracking": Uma matriz UBAR, de dimensão CON por N. Cada coluna tem os valores desejados do vetor de controle em um período. As colunas são para os períodos 0, 1, 2, ..., N-1. Valem as regras da Seção 1.3.15.

1.3.25: Uma matriz Z, de dimensão EX por N. Cada coluna tem os valores do vetor exógeno em um período. As colunas são para os períodos 0, 1, ..., N-1. Este ítem só aplica se o problema tiver vetores exógenos (EX > 0). Caso contrário, ítem 1.3.26 deve seguir item 1.3.24 imediatemente. Valem as regras da Seção 1.3.15.

1.3.26: O vector X[I,0] (I = 1, 2, ... ST) com os ST valores iniciais do vetor de estado.

1.4 Unidades ("units") do programa PQL:

1.4.0 Introdução:

A "unit" do Turbo Pascal 5.0 permite a divisão do programa em partes, ou unidades diferentes. As unidades são ligadas pela linha "USES" na seção inicial das unidades. O programa principal, PQL, chama as unidades na sequência lógica e pode ser considerado o coordenador das unidades. A divisão do programa em unidades facilita a sua organização, e aumenta a rapidez das operações de leitura, gravação, e compilação. O programa PQL tem dezessete unidades.

Em geral, o usuario não terá necessidade de mexer nas unidades. Mexer implica fazer modificações no programa ou corrigir "bugs". Mas, se realmente tiver que modificar o programa é necessario saber o papel das unidades.

Na maioria dos casos, a programação é simples, e aqui então é dado apenas uma breve descrição do trabalho da unidade. Porém, em outros casos o raciocínio por trás da programação não é tão evidente, e fornece-se um detalhamento mais amplo. Frequêntemente unidades diferentes têm rotinas semelhantes ou até idénticas. Neste caso, explica-se a rotina no texto de uma das unidades, e faz-se a referência nas outras unidades.

1.4.1 CRT:

A CRT é uma unidade pre-compilada que acompanha o Turbo Pascal 5.0. Ela permite várias operações na tela. Sendo pre-compilada, não pode ser alterada. É a única unidade pre-compilada do PQL.

1.4.2 CHANGE:

A CHANGE permite a modificação dos dados. Em geral, ela só será chamada pelo programa depóis do usuario ter escolhido a opção de fornecer os dados durante a execução do program (e não através de um arquivo de dados).

Quem digita dados no teclado provávelmente vai fazer erros. Como corrigir os erros? Uma maneira seria de exigir uma confirmação depóis de cada dado digitado. No entanto, considerou-se esta maneira tediosa. No seu lugar implementou-se o conjunto de unidades KEYDATA, VIEW e CHANGE. Na KEYDATA, os dados são digitados. Na VIEW, eles podem ser vistos. Na CHANGE, eles podem ser mudados.

Quando for escolhida a opção de fornecer os dados através do teclado, um arquivo de dados será posteriormente criado. Neste caso, se os dados forem mudados, o arquivo de dados terá os valores novos gravados no arquivo.

A CHANGE raramente será usada se os dados forem fornecidos por um arquivo de dados, porque é mas fácil fazer as mudanças no próprio arquivo. Porém, se a CHANGE for chamada, o usuario receberá a opção de gravar os novos dados naquele arquivo, sobre-escrevendo os dados originais. Vai aparecer a mensagem:

Deseja gravar as mudanças no arquivo original de dados (S/N ?)

Se a resposta for negativa, as mudanças já feitas só afetarão a execução atual do programa e não o arquivo de dados.

Além das variáveis da unidade GLOBAL, a CHANGE usa também as variáveis declaradas na MATVAR.

1.4.3 CRITVAL:

A CRITVAL tem dois procedimentos que calculam o valor da função objetivo. O procedimento Criterion_Value_Tracking trata o caso do problema de "tracking". Criterion_Value_General trata o caso geral. Nos dois casos, depóis de efetuar-se os cálculos o valor é impresso no arquivo de saída resumida.

Uma explicação da programação para multiplicar matrizes é fornecida na Seção 1.4.10, MATRIX.

As formas das funções objetivos podem ser vistas na Seção 1.1.

1.4.4 FEEDBACK:

A FEEDBACK usa os valores de G_k e g_k, e a regra de realimentação para determinar u_k. Então, com u_k e as equações de sistema, calcula-se x_{k+1}. A regra de realimentação tem a forma:

$$(5) \quad u_k = G_k x_k + g_k$$

Com os valores de x_0 (dado) e G_0 e g_0 (calculados pela unidade MATRIX), pode-se determinar u_0.

As equações de sistema têm a forma:

$$(6) \quad x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

Tendo A, x_0, B, u_0, c, D, e z_0, pode-se calcular x_1. Voltando à regra de realimentação, pode-se determinar u_1, e continuando, x_2.

Então, a FEEDBACK alterna a aplicação da regra de realimentação e as equações de sistema para calcular u_0, x_1, u_1, x_2, $\ldots u_{N-1}$, x_N.

Fornece-se uma explicação da programação para multiplicar matrizes na Seção 1.4.10, MATRIX.

Deriva-se a regra de realimentação no Capítulo 3.

1.4.5 FILEDATA:

A FILEDATA cria um arquivo de dados para dados entrados pelo teclado durante a execução do programa (e não por um arquivo de dados). Pede-se o nome deste arquivo quando o usuario escolhe a opção de digitar os dados. Então, para resolver o problema (ou uma variação) novamente no futuro não é necessário digitar novamente todos os dados. Se os dados forem lidos de um arquivo de dados, o program não chama a unidade FILEDATA.

1.4.6 GLOBAL:

Esta unidade declara as variáveis globais. As outras unidades chamam GLOBAL na linha USES, assim podendo usar estas variáveis.

Neste arquivo encontra-se práticamente a única excepção à regra geral de não mexer nas unidades. Pode-se alterar as constantes MAXCON, MAXST, MAXN, e MAXEX para fazer o programa acessível aos problemas com dimensões que excedem as para que ele esta configurado.

A memória no "stack" limita o tamanho do problema que pode ser resolvido usando PQL. A instalação original tem os seguintes limites nas dimensões das variáveis:

MAXCON	(controle) :	3
MAXST	(estado) :	28
MAXN	(períodos) :	22
MAXEX	(exogenous) :	5

Esta dimensionamento não deixa folga na memória, e para o programa executar apenas podem ser carregados na RAM o sistema operacional DOS, o Turbo Pascal, e o programa PQL. Programas residentes na memória RAM devem ser evitados, ou o dimensionamento deve ser modificado conforme as instruções a seguir.

Usam-se estas "constantes" no dimensionamento das matrizes usadas pelo programa. As diferenças nas dimensões das variáveis tentam refletir os tamanhos dos diferentes tipos de vetores na prática. Porém, se um problema não couber na memória as dimensões das variáveis podem ser modificadas para alocar memória de uma variável para outra.

Ainda pode-se resolver um problema com quatro elementos de controle, dado que existe folga nas dimensões das outras variáveis. Se a variável de estado tiver apenas 20 elementos, pode-se realocar a sua memória que sobra para a variável de controle. Para fazer isto, modifica-se o código de:

```
CONST
  MAXDIM = 28;
  MAXCON =  3;
  MAXST  = 28;
  MAXN   = 22;
  MAXEX  = 5;
```

Para:

```
CONST
  MAXDIM = 22;
  MAXCON =  4;
  MAXST  = 20;
  MAXN   = 22;
  MAXEX  = 5;
```

Se folga existir em outras variáveis, também deve-se realocá-la em favor da variável que excede a dimensão máxima. Então, deve-se re-complilar o programa para ver se a nova configuração cabe dentro da memória. A mensagem de erro "Structure too large" (estrutura grande demais) implica que o problema ainda não cabe. Neste caso, recomenda-se que o próprio modelo seja modificado. Talvez seja possível incluir vários elementos do vetor de estado num elemento só. Por exemplo, em lugar de ter elementos separados para "vendas de carros a alcool" e "vendas de carros a gasolina" pode-se falar de "vendas de carros". Semelhantemente, pode-se considerar a agregação de períodos de tempo, tal como usando dados anuais em vez de semestrais.

Note que deve-se mudar a constante MAXDIM para o máximo entre MAXCON, MAXST, MAXN, e MAXEX. O dimensionamento de algumas matrizes auxiliares é uma função do MAXDIM.

Finalmente, deve-se repetir a realocação feita no arquivo GLOBAL no arquivo do pacote gráfico, PLOTTER.PAS.

Deve-se também mencionar as constantes InWidth e OutWidth. InWidth (Seção 1.3.13) controla o número de colunas suposto na entrada de dados. O valor "default" é oito. OutWidth (Seção 1.3.6) controla o número de colunas nos arquivos de saída. O valor "default" é cinco.

1.4.7 GRDATA:

A GRDATA cria um arquivo de dados em uma forma para leitura e uso com o programa gráfico. Especifica-se o nome deste arquivo no arquivo de dados original. Quando o programa gráfico for executado, tem-se que digitar o nome deste arquivo.

1.4.8 INDATA:

A unidade INDATA extrai os dados do arquivo de entrada, o nome do qual o programa pergunta. Instruções para colocar os dados no arquivo de dados são apresentadas na Seção 1.3.

O procedimento principal, Read_Data, é pouco mais de que uma sequência de comandos READ. Para ler as matrizes, o PQL chama o procedimento Read_Data_File. Este procedimento determina o número de partições da matriz, que depende de quantas colunas a matriz tem. No arquivo de dados a largura máxima da linha de dados é de oito colunas (a constante InWidth, na unidade GLOBAL). Uma matriz de 20 colunas então deveria aparecer no arquivo de dados em três partições, as primeiras duas tendo oito colunas e a última com quatro. Então, Read_Data_File também calcula o número de colunas da última partição, que pode variar entre um e oito. Veja o exemplo na Seção 1.3.13.

INDATA verifica que as dimensões do problema estão dentro dos limites estabelecidos na unidade GLOBAL. Não verifica se os dados foram corretamente colocados no arquivo de entrada. Pode-se verificar isto usando a opção "Ver Dados" antes que o problema é resolvido. Alternativamente, a entrada correta das variáveis pode ser comprovada no arquivo de saída detalhada através da colocação de PrintSwitch[1] com valor 1.

As linhas ativas de INDATA dependem de se o problema é do tipo de "tracking" ou do tipo geral.

1.4.9 KEYDATA:

Quando executa-se o programa, ele chama ou INDATA ou KEYDATA. INDATA pega os dados de um arquivo de dados. KEYDATA oferece a opção de digitar os dados no instante. Este segundo método, em geral, é muito mais trabalhoso do que usar um arquivo de dados. Não é recomendado exceto para problemas muito pequenos.

O arquivo é auto-explicativo, e tem extensas instruções para orientar o usuario.

Em geral, o usuario não tem a oportunidade imediata de mudar o que ele entra. Isso pode ser feito posteriormente usando as opções "Ver Dados" e "Mudar Dados" das unidades VIEW (Seção 1.4.17) e CHANGE (Seção 1.4.2). Porém, o PQL pede confirmação imediata do número de elementos dos vetores exógenos (EX), de controle (CON), de estado (ST) e do número de períodos (N). As dimensões das matrizes a serem entradas dependem destes valores.

Note que quando Turbo Pascal espera um número e um caráter for digitado sem querer, um erro que termina o programa resulta. Perde-se todos os dados digitados até então. Isto dá uma boa razão para usar um arquivo de dados. Note também que a virgula é um caráter, e que não deve-se digitá-la como parte de um nûmero. Mil deve ser digitado "1000" e não "1,000".

Para algumas matrizes o usuário pode selecionar entre duas modas de entrada: densa ou esparsa. Para matrizes com muitos zeros a segunda opção é mais eficiente. O usuario digita as posições e os valores dos elementos não zeros, e, automaticamente, os outros são fixados em zero.

1.4.10 MATRIX:

A unidade MATRIX é a maior do programa. A maior parte dela consiste nas multiplicações de matrizes para calcular as matrizes Ricatti e as da equação de realimentação, G_k e g_k. O desenvolvimento matemático é apresentado no Capítulo 3.

Para multiplicar uma matriz A (N x M) por uma matriz B (M x P) tem-se a seguinte forma:

```
{Initializar a matriz}
FOR Ind1 := 1 TO N DO
    FOR Ind2 := 1 TO P DO
        AB[Ind1,Ind2] := 0;

{Multiplicar a matriz A pela matriz B}
FOR Ind3 := 1 TO M DO
    FOR Ind1 := 1 TO N DO
        FOR Ind2 := 1 TO P DO
           AB[Ind1,Ind2] := AB[Ind1,Ind2] + A[Ind1,Ind3]*B[Ind3,Ind2];
```

Dependendo nos valores dos "switches" de impressão, as matrizes intermediárias aparecem ou não no arquivo de saída detalhada. Se o usuario pediu impressão, uma unidade chamada PRROU (Seção 1.4.14) é executada. As matrizes intermediárias que podem ser impressas são listadas na Seção 1.3.2.

Além de multiplicar e imprimir matrizes a unidade MATRIX determina a inversa da matriz THETA no procedimento INVERT, que tem um outro procedimento embutido chamado "SWITCH_LINE".

Para determinar a inversa da matriz, usa-se o método Gauss-Jordan. Já que o significado do código não é auto-explicativo, aqui divida-se ele e o interpreta parte por parte. Depóis, para ilustrar o andamento do procedimento, fornece-se á primeira iteração de um exemplo.

PASSO 1: Cria uma matriz auxiliar AUX (CON x 2*CON). A primeira partição desta matriz (coluna 1 até CON) consiste na própria THETA. A segunda partição é a matriz unidade.

```
{Initializar matriz AUX}

FOR Ind1 := 1 TO CON DO
    FOR Ind2 := 1 to 2*CON DO
        AUX[Ind1,Ind2] := 0;

FOR Ind1 := 1 TO CON DO
    FOR Ind2 := 1 TO CON DO
        AUX[Ind1,Ind2] := THETA[Ind1,Ind2];

FOR Ind1 := 1 TO CON DO
    AUX[Ind1,CON+Ind1] := 1;
```

PASSO 2: Fazer L = 1. Usa-se este índice para indentificar a linha e a coluna atual.

```
L := 1;
```

PASSO 3: Verifica que existe pelo menos um elemento diferente de zero na coluna ativa, ou na linha ativa ou abaixo dela. Caso contrario, a matriz não é inversível e o programa termina.

```
INV := FALSE;
FOR Ind1:= L TO CON DO
    IF ABS(AUX[Ind1,L]) - Tol >= 0 THEN
        INV := TRUE;

IF INV = FALSE THEN
    BEGIN
        WriteLn;
        WriteLn('A Matriz THETA ',T,' nao e inversivel');
        WriteLn('Toque qualquer tecla para terminar o programa.');
        Repeat Until Keypressed;
    END;
```

PASSO 4: Substitua na linha ativa a linha que tem o maior valor absoluto na coluna ativa. O procedimento SWITCH_LINE faz esta troca de linhas. A nova linha ativa será ou a original ou uma linha anteriormente abaixo da original. A interseção da linha e a coluna ativa é chamada o pivô.

```
FOR Ind1 := L+1 TO CON DO
    IF ABS(AUX[L,L]) - ABS(AUX[Ind1,L]) < 0 THEN
        SWITCH_LINE;
```

PASSO 5: Dividir todos os elementos da linha ativa pelo valor do elemento pivô. Note que o programa não divide os elementos na matriz na linha ativa AUX antes do pivô porque eles já teriam sido zerados.

```
PIV := AUX[L,L];

FOR Ind3 := L TO 2*CON DO
    AUX[L,Ind3] := AUX[L,Ind3]/PIV;
```

PASSO 6: Para todas as linhas não-ativas, faz a seguinte operação: substitua o valor atual de cada elemento por seu valor menos o produto do elemento da coluna do pivô na mesma linha e o elemento da mesma coluna na linha ativa.

```
FOR Ind1:=1 TO CON DO
    IF Ind1 <> L THEN
        BEGIN
            TEMP := AUX[Ind1,L];
            FOR Ind2 := 1 TO 2*CON DO
                AUX[Ind1,Ind2] := AUX[Ind1,Ind2] - AUX[L,Ind2]*TEMP;
        END;
```

PASSO 7: Fazer L = L + 1;

```
L := L + 1;
```

PASSO 8: Se L > CON: --> PASSO 3.

Caso Contrario: --> PASSO 9.

PASSO 9: Fazer THETA_INV = AUX.

```
FOR Ind1:=1 TO CON DO
    FOR Ind2:=1 TO CON DO
        THETA_INV[Ind1,Ind2] := AUX[Ind1,CON + Ind2]
```

Suponha que quer-se invertir a matriz THETA (CON x CON)

Primeira Iteração:

$$\mathrm{THETA} = \begin{vmatrix} 1 & 2 & 0 \\ 4 & 10 & 4 \\ 3 & 10 & 9 \end{vmatrix}$$

PASSO 1:

$$AUX = \left| \begin{array}{ccc|ccc} 1 & 2 & 0 & 1 & 0 & 0 \\ 4 & 10 & 4 & 0 & 1 & 0 \\ 3 & 10 & 9 & 0 & 0 & 1 \end{array} \right|$$

PASSO 4:

$$AUX = \left| \begin{array}{ccc|ccc} 4 & 10 & 4 & 0 & 1 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 \\ 3 & 10 & 9 & 0 & 0 & 1 \end{array} \right|$$

PASSO 5:

$$AUX = \left| \begin{array}{ccc|ccc} 1 & 2.5 & 1 & 0 & .25 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 \\ 3 & 10 & 9 & 0 & 0 & 1 \end{array} \right|$$

PASSO 6:

$$AUX = \left| \begin{array}{ccc|ccc} 1 & 2.5 & 1 & 0 & .25 & 0 \\ 0 & -.5 & -1 & 1 & -.25 & 0 \\ 0 & 2.5 & 6 & 0 & -.75 & 1 \end{array} \right|$$

No passo 3, a inversibilidade da matriz THETA é testada. Um valor menos do valor da variável Tol é considerado 0. Na instalação original do pacote, Tol é igual a 0.0000000005. Este valor deve ser adequado para problemas usando variáveis do tipo REAL. Se um problema precisar de números muito pequenos, as declarações das variáveis poderiam ser mudadas para os tipos DOUBLE ou EXTENDED, e o valor da Tol reduzido.

1.4.11 MATVAR:

A MATVAR declara um conjunto de variáveis "semi-globais", variáveis que usa-se em algumas, mas não todas as unidades. Estas variáveis não estão incluídas na unidade GLOBAL para conservar memória, assim permitindo problemas de maior dimensão a serem resolvidos.

1.4.12 MULTIPLY:

O program chama a unidade MULTIPLY quando solicitou-se os multiplicadores através dos PrintSwitches 12 até 15. Para uma explicação do significado dos multiplicadores, veja Seção 2.6.

PrintSwitch[12] refere à matriz dos multiplicadores de impacto. Esta matriz é a mesma matriz B, e a sua determinação não precisa de cálculos.

PrintSwitch[13] refere às matrizes dos multiplicadores defasados.

PrintSwitch[14] refere às matrizes dos multiplicadores de prazo intermediário. No cálculo dos multiplicadores de prazo intermediário usa-se os multiplicadores defasados.

PrintSwitch[15] refere à matriz de multiplicadores de muito longo prazo, que (diferente das outras matrizes de multiplicadores) pode ou não existir. Se não existir, a matriz $(I-A)^{-1}$ não é inversível e uma mensagem correspondente aparecerá no arquivo de saída.

A MULTIPLY tem três outros tipos de rotinas que merecem mais explicação. Discutem-se a inversão e multiplicação de matrizes na Seção 1.4.10, MATRIX. Explica-se a rotina de impressão no arquivo de saída na Seção 1.4.8, INDATA.

1.4.13 OUTDATA:

A OUTDATA é chamada quando o usuario quer que o arquivo de saída detalhada contenha os dados de entrada (PrintSwitch[1] = 1). PQL fixa PrintSwitch[1] = 1 quando os dados são digitados durante a execução do programa. Neste caso o usuario não pode suprimir a impressão dos dados. Como é o caso em outras unidades, um procedimento que imprime as matrizes destes dados faz partições delas. Se chama o procedimento repetitivamente para as várias matrizes.

1.4.14 PRROU:

A unidade PRROU é uma coleção de procedimentos para imprimir as matrizes intermediárias. Chama-se PRROU só da unidade MATRIX.

1.4.15 QUADFORM:

So é chamada a unidade QUADFORM quando o problema é do tipo "tracking". A QUADFORM calcula os vetores de penalidade w_{-k} e l_{-k}, assim permitindo o uso do algoritmo derivado para o caso geral. Veja Capítulo 2.

A formulas para calcular w_{-k} e l_{-k} são:

$$(7)\quad w_{-k} = -\bar{W}_k x_k \qquad k = 0, 1, \ldots, N$$

$$(8)\quad l_{-k} = -\bar{L}_k u_k \qquad k = 0, 1, \ldots, N-1$$

Note que no problema geral, há a simplificação de definir l_{-k} igual para todos os valores de k. Pode-se dizer o mesmo com respeito a w_{-k}, para todos os valores de k exceto k = N. No caso de "tracking", $w_{-k} \neq w_{-k+1}$ e $l_k \neq l_{k+1}$.

1.4.16 SUMMARY:

A unidade SUMMARY contem dois procedimentos, cada um dos quais cria um arquivo que mostra os vetores de estado e controle da solução. O Write_Summary_Tracking é chamado se o problema for de "tracking". Além dos valores calculados, também mostram-se os valores desajados e a diferença entre os dois. O Write_Summary é chamado se o problema for geral.

1.4.17 VIEW:

A unidade VIEW permite que os dados lidos pelo programa possam ser vistos pelo usuario. Assim, VIEW facilita o descobrimento de dados incorretamente colocados.

Dado o tamanho limitado da tela, as matrizes grandes têm que ser partidas para serem mostradas. A partição é tanto vertical quanto horizontal.

ColBlocks determina o número de partições necessário para mostrar todas as colunas, e RowBlocks o número necessário para todas as linhas. O número de telas para mostrar a matriz é o produto de ColBlocks e RowBlocks.

Exemplo:

Matriz A é 28 por 28.

O número de linhas na tela é 20.

O número de colunas mostradas na tela é 5.

ColBlocks = TRUNC(28/5) + 1 = 6

RowBlocks = TRUNC(28/20) + 1 = 2

Partição	Linhas	Colunas
1	1 .. 20	1 .. 5
2	1 .. 20	6 .. 10
3	1 .. 20	11 .. 15
4	1 .. 20	16 .. 20
5	1 .. 20	21 .. 25
6	1 .. 20	26 .. 28
7	21 .. 28	1 .. 5
8	21 .. 28	6 .. 10
9	21 .. 28	11 .. 15
10	21 .. 28	16 .. 20
11	21 .. 28	21 .. 25
12	21 .. 28	26 .. 28

As colunas e as linhas aparecem numeradas na tela para facilitar a identificação dos elementos da matriz.

Inicialmente usou-se um procedimento geral para todas as matrizes. Porém, apareceram problemas de "overflow" no "stack". Tais problemas desapareceram quando foi dado a cada matriz a sua própria conjunto de instruções para partição.

Além da unidade GLOBAL, a VIEW também usa as variáveis listadas na unidade MATVAR.

1.5 O programa PLOTTER:

1.5.0 Introdução:

O programa PLOTTER é executado independentemente do PQL, mas usa-se os dados que o PQL fornece. O sétimo ítem do arquivo de dados do PQL contem o nome do arquivo de dados que cria-se para receber os dados para PLOTTER. A forma destes dados não é conveniente para a inspeção do usuario, mas sim para leitura do PLOTTER.

O PLOTTER isola um elemento do vetor de estado ou de controle e mostra os seus valores através do tempo. Pode-se mostrar em um gráfico até quatro resultados do PQL. Se o problema for do tipo "tracking", também mostram-se os valores desejados do elemento. Nota-se que estes valores alvos devem ser iguais para todas as execuções.

A PLOTTER só permite a examinação dos gráficos na tela, e não a sua impressão. Desenhou-se o programa para uso com um vídeo tipo Hercules (resolução 720 x 350). Funciona com qualquer outro tipo de vídeo, mas o seu desempenho é melhor com Hercules.

1.5.1 Usando o PLOTTER:

Executar o PLOTTER é fácil e pode ser entendido seguindo as instruções que aparecem na tela. Só precisa-se saber de antemão os nomes dos arquivos de dados gráficos a serem usados.

O programa pede primeiro o número de execuções de PQL a ser visto. Depóis disso, o PLOTTER pede os nomes dos arquivos de dados gráficos correspondentes. Tais nomes têm que ser confirmados depóis de serem digitados. Então, aparece o menu principal, que oferece as seguintes opções:

```
Mostrar Grafico de
1. Variavel de Estado
2. Variavel de Controle
3. Sair de programa de graficos
```

Depóis de selecionar um dos primeiros dois ítens, o PLOTTER pede o número do elemento no vetor a ser grafado. O intervalo dos elementos válidos é dado. Para um vetor de estado de oito elementos, a mensagem aparece assim:

```
Mostrar grafico de elemento de estado numero:
Digite numero entre 1 e 8:
Selecao?
```

Se a seleção for dentro do intervalo permitido, o gráfico aparece no vídeo e permanece alí até que uma tecla seja teclado. Aparece novamente o menu principal.

1.5.2: Descrição do programa:

Os comandos gráficos do PLOTTER representam uma extensão do Turbo Pascal além do que foi programado no PQL. Quem não conhece os comandos gráficos deve consultar um livro de Pascal antes de modificar este programa.

A unidade pre-compilada GRAPH do Turbo Pascal permite o usuario a isolar cada um dos pontos pequenos dos quais a tela é formada. Pode-se apagar ou ligar tais pontos, em várias cores no caso dos vídeos coloridos. Cada ponto da tela tem um endereço de duas coordenadas. A primeira é a distância horizontal do canto superior esquerdo, e a segunda é a distancia vertical do mesmo lugar. Note que não usa-se a convenção de colocar o ponto (0,0) no canto esquerdo inferior.

Considera o comando Circle, que coloca um círculo na tela. Os parâmetros do comando são o raio do círculo, e as suas coordenadas horizontais e verticais. Circle(3,50,100) coloca um círculo de raio três, com centro cinquênta pontos abaixo e cem pontos para a direita do canto superior esquerdo.

Infelizmente, este círculo aparece em lugares diferentes em vídeos diferentes. Os vídeos variam nas suas resoluções, e consequêntemente no número e espaçamento dos pontos. Então, um programa que usa endereços absolutos não será compatível com outros micros com outras resoluções.

Para melhorar a compatibilidade do PLOTTER, expressa-se todos os endereços em termos relativos às dimensões do vídeo que está sendo usado, que o Turbo Pascal detecta. Em PLOTTER, esta informação está contida nas variáveis xMax e yMax. Mesmo com tudo expresso em termos relativos, o programa ainda funciona melhor nos vídeos de alta resolução.

O programa é constituido por um programa principal e vários procedimentos. O programa principal pede os nomes dos arquivos de dados a serem usados, e lê os mesmos. Então, chama o menu principal, mostrada na Seção 1.5.1.

O procedimento a ser chamado próximo depende da seleção do menu. Porém, os procedimentos para mostrar gráficos de elementos de controle têm as mesmas estruturas que aqueles de elementos de estado. Os comentários a serem feitos com respeito ao procedimento STATE também aplicam ao procedimento CONTROL.

Primeiro, o procedimento pergunta o número do elemento de estado a ser grafado, verificando que a resposta é um elemento válido. Então, para determinar o intervalo do eixo vertical os valores máximos e mínimos são determinados. No próximo passo, o procedimento Prepare_Graph determina as dimensões do vídeo sendo usado, e coloca os eixos e os seus rótulos na tela.

No resto do STATE coloca-se os pontos correspondentes aos valores de X através do tempo. Se o problema for do tipo "tracking", os valores de XBAR também são colocados. Ligam-se os pontos como linhas. A coordenada vertical de um ponto é uma função do seu valor, e a coordenada horizontal é uma função de tempo. Por isso, a separação horizontal entre pontos consecutivos é constante. Depóis de colocar os pontos, o PLOTTER coloca abaixo do eixo horizontal os rótulos que identificam os pontos com as suas execuções correspondentes.

O símbolo que marca um ponto depende da variável e da execução. Um ponto de XBAR é indicado por um "+". X na primeira execução é marcado por um círculo oco. Se tiverem execuções subsequentes, na segunda o X é marcado por um círculo cheio, na terceira por um elipse vertical, e na quarta por um elipse horizontal.

O gráfico permanece no vídeo até que uma tecla seja tocado. Então, volta novamente ao menu principal.

APÊNDICE II: ALGORITMO DE SOLUÇÃO PARA O PROBLEMA QUADRÁTICO LINEAR DETERMINÍSTICO DISCRETO DE CONTROLE ÓTIMO

I. Forma geral do problema:

O problema quadrática linear determinístico discreto de controle ótimo tem a seguinte forma:

Achar u_k para $k = 0, 1, \ldots, N-1$

$$\text{Min } J = \tfrac{1}{2}x_N' W_N x_N + w_N' x_N + \sum_{k=0}^{N-1} \left(\tfrac{1}{2}x_k' W_k x_k + w_k' x_k + \tfrac{1}{2}u_k' L_k u_k + l_k' u_k\right)$$

Sujeito a:

$$x_{k+1} = Ax_k + Bu_k + C + Dz_k$$

para $k = 0, 1, \ldots, N-1$

com x_0 dado

II. Algoritmo de solução:

1. Calcular iterativamente as matrizes da regra de realimentação e as matrizes de Ricatti, começando com o penúltimo período e progredindo de trás para frente.

Matrizes da regra de realimentação:

$$G_k = -(THETA_k)^{-1} PSI_k$$

$$g_k = -(THETA_k)^{-1} theta_k$$

Matrizes de Ricatti:

$$K_k = PSI_k + G_k THETA_k G_k + 2 \cdot PSI_k G_k$$

$$p_k = G_k' THETA_k g_k + PSI_k g_k + phi_k + G_k' theta_k$$

Onde:

$$PHI_k = A'K_{k+1}A + W_k$$

$$THETA_k = B'K_{k+1}B + L_k$$

$$PSI_k = A'K_{k+1}B$$

$$phi_k = A'K_{k+1}C + A'K_{k+1}Dz_k + A'p_{k+1} + w_k$$

$$theta_k = B'K_{k+1}C + B'K_{k+1}Dz_k + B'p_{k+1} + l_k$$

$$mu_k = \tfrac{1}{2}C'K_{k+1}C + z_k'D'K_{k+1}C + \tfrac{1}{2}z_k'D'K_{k+1}Dz_k + p_{k+1}'C + p_{k+1}'Dz_k$$

Note-se que as matrizes Ricatti são as matrizes de penalidade no período final:

$$K_N = W_N$$

$$p_N = w_N$$

2. Aplicar iterativamente a regra de realimentação e as equações de sistema para determinar os vetores u_k e x_k. Os cálculos devem começar com o vetor u_0 e continuar até calcular x_N.

Regra de realimentação:

$$u_k = G_k x_k + g_k$$

Equações de sistema:

$$x_{k+1} = Ax_k + Bu_k + C + Dz_k$$